

统计分析方法丛书

YINGYONG
STATA ZUO
TONGJI FENXI

第5版

应用STATA做统计分析

劳伦斯·汉密尔顿 著
郭志刚 等 译



重庆大学出版社
http://www.cqup.com.cn

INGYONG STATA ZUO TONGJI FENXI

本书引导读者将统计方法与统计软件 STATA 联系起来,达到迅速学会应用统计方法做研究的目的。

本书从STATA软件与STATA的资源，数据管理，制图，概要统计及交互表，方差分析和其他比较方法，线性回归分析，回归诊断，拟合曲线，稳健回归，LOGISTIC回归，生存模型与事件计数模型，主成分、因子和聚类分析，时间序列分析，编程入门，等等，完整而精练地介绍了STATA软件或软件包的各项基本功能和在统计分析中的应用。全书以列举实例的方式编写，并穿插了上百幅图片，广泛引证各种相关资料中的数据，简明地介绍了常用的各种命令的分析运行情况，便于学习掌握。此外，在最后一章拓展性地介绍了常用的编程知识和技能，以便于能更加灵活地运用STATA软件做更多的统计分析。

本书突出了程序性、实用性、完整性，本书兼具教材和使用手册的特点，适宜作为致力于统计学研究和数据分析应用的专家和学者自学参考。

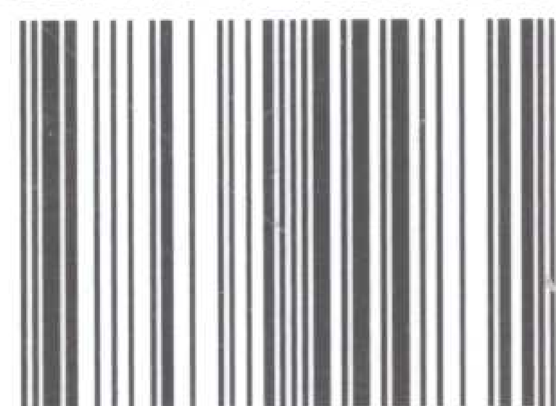
本书示例数据通过以下途径可以下载：

http://q.blog.sina.com.cn./fafang(万卷方法与学术规范博客圈)

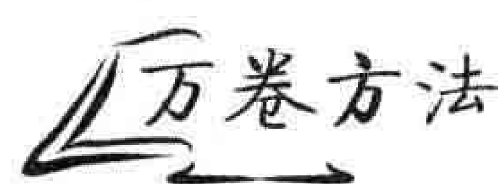
http://www.cqup.cn/edusrc/TeachingDetail.aspx?ID=4483

http://www.brookscole.com/cgi-wadsworth/course_products_wp.pl?fid=M20b&flag=student & product_isbn=9780495109723&discipline_number=17

ISBN 978-7-5624-4483-1



定价: 58.00元



统计分析方法丛书

YINGYONG
STATA ZUO
TONGJIFENXI

第5版

应用STATA做统计分析

劳伦斯·汉密尔顿 著
郭志刚等 译



重庆大学出版社

图书在版编目(CIP)数据

应用 STATA 做统计分析/(美)汉密尔顿(Hamilton, L)著;郭志刚等译.—重庆:重庆大学出版社,2008.8

(万卷方法.统计分析方法丛书)

书名原文:Statistics with Stata

ISBN 978-7-5624-4483-1

Ⅰ.应… Ⅱ.汉… Ⅲ.郭… Ⅳ.统计分析—应用软件,STATA
Ⅴ.C812

中国版本图书馆 CIP 数据核字(2008)第 119370 号

应用 STATA 做统计分析

劳伦斯·汉密尔顿 著

郭志刚 等 译

责任编辑:张立武 版式设计:雷少波

责任校对:邹 忌 责任印制:赵 晟

*

重庆大学出版社出版发行

出版人:张鸽盛

社址:重庆市沙坪坝正街 174 号重庆大学(A 区)内

邮编:400030

电话:(023) 65102378 65105781

传真:(023) 65103686 65105565

网址:http://www.cqup.com.cn

邮箱:fxk@cqup.com.cn (市场营销部)

全国新华书店经销

重庆升光电力印务有限公司印刷

*

开本:787×1092 1/16 印张:23.25 字数:574 千 插页:16 开 2 页

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷

印数:1—3 000

ISBN 978-7-5624-4483-1 定价:58.00 元

作者前言

非常高兴《应用 Stata 做统计分析》一书首次译成中文版，通过郭志刚教授的辛勤工作，该书翻译得相当专业。自 1990 年以来，本书的英文版经过了多次修订和再版，已经非常成功。为了顺应 Stata 自身的发展，随着每一版修订，《应用 Stata 做统计分析》变得越来越丰富，并且覆盖了越来越多的主题。当郭志刚教授与我联系出版中译本时，我为能有更多的新读者研读本书而感到高兴。

1990 年，当我写作首版《应用 Stata 做统计分析》时，Stata 刚刚推出，但已经令人振奋。Stata 逻辑一致的命令和能使用不同工具进行工作的便捷之处都使得这个软件优越于当时的竞争对手，并成为现代桌上电脑时代的理想选择。比如，它可以在简单表格、高级模型和图形之间迅速切换，以取得对数据的理解。多年来，Stata 已经非常成熟。多次更新升级，增加了大量的新特色，从用户友好的菜单和环球网链接到高级模型和编程。Stata 的内置编程语言被证明非常重要，因为它允许用户编写自己的程序。因此，发表在 Stata 期刊中的新技术可以迅速被 Stata 接纳，有时这些新技术就是用 Stata 来进行首创研制的。与此同时，Stata 仍在继续增加其基本统计的功能和对复杂数据库管理的支持。Stata 的稳步发展证明了它从一开始就有极佳的设计。尤其值得说明的是，虽然技术上具有巨大进步，但 Stata 仍然比其他统计软件更容易学。

《应用 Stata 做统计分析》是关于 Stata 的第一本书。与该软件本身一样，本书的目的在于填补一些空白。我是为学生和实际研究者写作本书的，希望在侧重理论的教科书和数千页内容的 Stata 手册之间架起桥梁。现代研究者需要各种各样的技术来分析他们的数据。因此，《应用 Stata 做统计分析》从基础的题目谈起，比如，统计学基础内容或者如何建立一个新数据集等。然后，再进一步介绍那些中级和高级主题，比如，回归诊断、logit 模型、稳健回归、因子分析、生存分析、时间序列模型以及编程工作。其中一些问题可能出现在研究生统计教学中，而另一些问题则可能会在研究中碰到。在每一章中，我都尤其关注两个实践问题：一个是"我如何用 Stata 做这一分析"，另一个是"这些统计结果能告诉我什么"。我的目的是写一本读者在工作时想摆在计算机旁的书。感谢郭志刚教授的翻译，中国读者现在有机会自己来判断它是否有用了。

劳伦斯·汉密尔顿(Lawrence Hamilton)

2007 年 12 月

译者前言

我接触 Stata 软件虽然还算比较早，但在很长时间内却主要是在用其他的统计软件，如 SYSTAT 和 SPSS。但是 2000 年以后，已经深感软件分析功能的局限，同时也为了给研究生开设更多高级社会统计学新课，我开始转向 Stata 软件。这一转向充分得益于劳伦斯·汉密尔顿教授的《应用 Stata 做统计分析》一书，尤其是在应用稳健回归、波松回归和生存分析等方面。由于该书注重实用，而 Stata 的命令语法又比较简单，所以很容易应用于统计教学和研究。2006 年，重庆大学出版社雷少波编辑为万卷方法——一套目前在国内社会科学界已颇具影响的研究方法图书——征求选材意见时，我特意推荐了此书，并且承担了组织翻译此书的任务。

中文翻译工作得到了作者汉密尔顿教授的大力支持，他不仅慷慨地提供了原作的电子文档，还就我们翻译中发现的 Stata 第 9 版新作的更新和原书中存在的一些印刷错误作了一些修订，特别是重写了变化较大的第 12 章。这些支持和指导都使得本书中译本质量大大提高。

中译本的翻译工作是由我本人和我的三名社会统计专业博士研究生完成的。我历来将专业文献翻译作为研究生学习与训练的重要途径。翻译一个文献虽然会耗费很大力气，但要比仅仅阅读一个文献的收益大得多。这是因为翻译工作要求对每一个概念、每一段文本和每一步操作都要认真领会，否则就肯定翻译不好。

事先，由我建立了本书关键词的中英文对照表，一方面是为了全书翻译上的统一，另一方面也是为了最后能够形成中译本的关键词索引。英文教材的关键词索引是学习时的一种方便途径，由于统计学是一个体系，很多基本概念和方法会在不同章节中反复出现，因此关键词索引表可以使这些基本概念、方法在一本书内各处的应用一览无余。相比之下，中文教材不太重视这个环节则是一个很大的不足。

本书翻译工作分工为：郭志刚承担第 1,9,10,11,13,14 章；巫锡炜承担第 3,7,8,12 章；赵联飞承担第 5,6 章；焦开山承担第 4 章；郭志刚、巫锡炜承担第 2 章；由郭志刚和巫锡炜完成关键词索引表的制作；王军、李丁两位同学参与了后期通读清样工作。

最后，我对全书译稿进行了校对、修订和统稿。这个工作既是为了保证翻译质量，也是我与学生之间的教学交流形式。学生们通过翻译稿中的修订部分可以学到很多东西，当然我也从学生的翻译中学到很多东西。我们都深感受益于汉密尔顿教授的这本既有理论又有应用操作的好书，其中的很多示例真是引人入胜。作为最终成果的中译本，我们很乐意奉献给广大读者来分享。

由于译者水平有限，翻译中难免有不当之处，恳请读者指教。

郭志刚
2007 年 12 月于北京大学

英文版前言

《应用 Stata 做统计分析》一书旨在为学生和实际研究工作者在统计教材和 Stata 应用之间架设桥梁，以缩小两者之间的差距。作为这样一个中介角色，本书既不准备对某一合适教材作详细说明，也不打算描述 Stata 的全部特征。相反，本书示范了如何使用 Stata 来完成各种各样的统计任务。每章的题目遵循统计学概念主题展开，而不是只集中在特定的 Stata 命令上，这使得《应用 Stata 做统计分析》一书又具有与 Stata 参考手册不同的结构。比如，数据管理这章涉及了创建、更新数据文件以及改变数据文件结构的各种程序。在各章中，概要统计及交互表、方差分析和其他比较方法以及拟合曲线这几章也都包含许多不同技术在内而又具有类似性的宽泛主题。

前 6 章（直到常规最小二乘法回归）为一般性主题，大体上对应了应用统计学的基础课程，但是增加了深度，讨论了分析人员经常碰到的实际问题。比如，如何汇总数据、创建虚拟变量、绘制符合发表质量要求的图形或者如何将方差分析转换为回归形式。在第 7 章的回归诊断及随后各章中，我们转入到高级课程或原创研究的领域。这里，读者能够找到有关如何获得并解释诊断统计量和图形的基本信息与示范说明：如何进行稳健回归、分位数回归、非线性回归、logit 回归、序次 logit 回归、多类 logit 回归或者泊松回归；如何拟合存活时间和事件计数模型；如何通过因子分析和主成分分析来建构合成变量；如何将观测案例按经验类型或聚类作划分；如何对时间序列作图或建模。Stata 近年来一直致力于与时俱进以保持其领先水平，并且这种努力尤其体现于它所提供的广泛的回归和模型拟合命令。

最后，我们以 Stata 编程的简介来结束全书。许多读者将会发现 Stata 可以做他们想做的任何事情，因此他们不需要编写原始程序。但是，对于积极主动的少数人而言，编程能力也是 Stata 的主要吸引力之一，并且它也肯定构成了 Stata 广泛传播和快速发展的基础。这一章为想学 Stata 编程的初学者开启了大门，不论是用于专门的数据管理，还是建立一种新的统计方法，是进行蒙特卡罗实验，还是为了教学。

通常，对于 Windows、Macintosh 和 Unix 等操作系统的计算机都有类似版本（“口味”）的 Stata 可以安装运行。跨越所有的操作系统，Stata 都使用相同的命令、数据文件和输出。这些版本只是在屏幕外观、菜单和文件处理的一些细节上有些差异，这是因为 Stata 需要遵循每一操作系统的规则。比如，在 Windows 系统下采用诸如“\目录\文件名”这样的文件设定，而在 Unix 系统下则采用“/目录/文件名”的设定。我并未示范所有三种规则，而是只采用了 Windows 规则，但是采用其他操作系统的用户应能发现，其实只需要稍加改变即可应用。

关于第5版的说明

我从1985年开始使用 Stata,当时还是它的首次发布(Stata 在2005年迎来了它的20周年纪念,为此该年 Stata 杂志的第5辑第1期(Stata Journal, 2005:5(1))开辟了一期特刊,刊登有关 Stata 发展史的文章和访谈,其中也包括《应用 Stata 做统计分析》一书的简史)。起初,Stata 只能在 MS-DOS 系统的个人电脑上运行,但是其 PC 取向使得它明显比其主要竞争对手更为现代,因为那时大多数竞争者还处于桌面革命之前的取向,还基于主机环境、使用80列穿孔卡的 Fortran 语言。与认为每个用户都是一堆卡片的主机统计软件不同,Stata 将视为与用户的一次对话。它的互动性以及统计程序与数据管理和制图的浑然一体,支持了分析思维的自然流程,而这些方面则是其他程序所不具备的。**graph**(作图命令)和 **predict**(预测命令)很快成为备受欢迎的命令。因此,我开始写作《应用 Stata 做统计分析》的最初版本,对应着 Stata 第2版,并于1989年出版。

该书问世以来,Stata 已经发生了巨大变化。我在该书中就注意到,"Stata 并不是一个万能程序……但是只要是它做,它就做得棒极了"。Stata 性能的扩展一直都引人注目。这一点在模型拟合程序的激增以及随后不断条理化方面显而易见。William Gould 为 Stata 建立的架构,包括其编程工具和统一的命令语法都已非常成熟,并已证明能够容纳新发展出来的统计思想。第10章开头提供的大量建模命令或第2章中的庞大函数清单都说明,多年来 Stata 在这些方面日益丰富。比如,适用于面板数据(**xt**)、调查数据(**svy**)、时间序列数据(**ts**)或存活时间数据(**st**)等方面的套装新技术开辟了更多可能领域,像非线性回归(**nl**)和一般化线性模型(**glm**)以及最大似然估计的一般程序的可编程命令也同样做到了这点。其他的关键性扩展还包括了矩阵编程能力的发展和新的数据管理性能。在最初版本的《应用 Stata 做统计分析》中,数据管理只是一个附带的题目;但是在这次第5版中,数据管理自然也就理所应当地成为篇幅上位居第二大的一章。

Stata 第8版标志着其发展史上最为根本性的升级,这一提升由新的菜单系统或 GUI(图形用户界面)和完全重新设计的制图能力所引领。发展于学生版程序 StataQuest 的一个有限的菜单系统自第4版以来就已经是可获得的选择了,而 Stata 第8版则首次装备了一个整合的菜单界面,包括了整套类型化命令选择。通过探索就可以学习这些菜单,这远比读一本书来得更容易,因此《应用 Stata 做统计分析》只在每一章的开始提供一些有关菜单的一般性建议。在绝大部分篇幅中,本书都采用命令方式来展示 Stata 能做什么;而这些命令的相应菜单应当能很容易地找到。

Stata 第8版重新设计的制图能力要求在第3章中作相应的大幅修改,因此它就成为本版中篇幅最长的一章。这个题目本身就很复杂,正如厚厚的《图形参考手册》(*Graphics Reference Manual*)(以及散见在文档中的其他材料)所表明的那样。我并不打算对基于命令的参考手册加以浓缩,而是采用了一种完全不同的、以实例为依据的补充方式。因此,

第3章提供了49幅各种各样图形的系统化图库,并且提供了每一图形如何绘制的说明。更多的实例则贯穿于全书的各个部分,甚至连第14章中的最后一些图形也示范了新的变化。因此,《应用 Stata 做统计分析》在一定程度上出乎意料地变成了新图形的展示柜。

相对于先前的版本(第5版),值得注意的变化是:包括了一些新的章节,诸如面板数据(第6章)、稳健标准误(第9章)和聚类分析(第12章)等。若干章中的命令示范一节也得到了修正和扩充。那些阅读过老版《应用 Stata 5 做统计分析》的读者还将发现一些更多的变化,包括新强调了对网络资源的利用(第1章),介绍了数据管理工具(第2章)、制表命令(第4章)、方差分析图形(第5章)、对多元共线性更敏锐的考察(第8章)、基于中位数的稳健的类方差分析(第9章)、用于多分类 logit 模型的条件效应标绘图(第10章)、一般化线性模型(第11章),还有一章全新的时间序列分析(第13章)以及重写的关于编程的一章(第14章)。关于 Stata 的其他新面貌或对旧命令的改进(包括 **graph** 和 **predict** 两个命令)也散见在书中各个部分。由于 Stata 现在可以做的事情太多,远远超出一本介绍性图书所能涵盖的范围,因此本版《应用 Stata 做统计分析》只是在绝大部分章开始的"命令示范"节中简要地展示了更多的程序,或者在选项清单之后提示了如何用 **help** 命令来查询有关细节。Stata 在线帮助和搜索功能也与程序同样升级换代和更新,因此这些也都是有益的建议。

除了这些帮助文件以外,可用资源还包括了 Stata 的网站、互联网及其文献搜索能力、用户群清单管理程序、网络课程、Stata 杂志,以及 Stata 大量的印刷文献(目前已超过5 000页,而且还在增加)。《应用 Stata 做统计分析》提供了 Stata 的便捷入门,而原文有"这些"其他资源则能够提供更多的帮助。

致 谢

Stata 的设计师 William Gould 值得受到称赞,因为是他创建了《应用 Stata 做统计分析》所描述的这个一流程序。我的编辑 Curt Hinrichs,在一些审稿人和忠实读者的支持下,说服我开始为写作一本新书而努力。从一开始,Stata 公司的 Pat Branton,以及 Shannon Driver、Lisa Gilmore 等许多专家学者,常常在很短期限内给出非常宝贵的反馈。如果没有他们的帮助,这一努力是不可能的。Alen Riley 和 Vince Wiggins 对我有关图形的问题迅速地作出回应。James Hamilton 贡献了有关时间序列的建议。Leslie Hamilton 校对了最终的手稿。

本书是围绕着数据分析而写就的。为了避免无休止地循环使用我自己的旧例,我使用了其他人著作中的例子,包括 Carole Seyfrit、Sally Ward、Heather Turner、Rasmus Ole Rasmussen、Erich Buch、Paul Mayewski、Loren D. Meeker 和 Dave Hamilton。Steve Selvin 分享了他的数个来自《应用生物统计方法》中第11章的例子。书中这些例子的数据来源于一些机构,包括冰岛统计局、格陵兰统计局、加拿大统计局、西北大西洋渔业组织、格陵兰自然资源研究所和加拿大渔业和海洋部。Brenda Topliss 所作的一次演讲激发了第14章中"gossip"编程的例子。其他形式的鼓励或思路则来自于 Anna Kerttula、Richard Haedrich、Jeffery Runge、Igor Belkin、James Morison、Oddmund Otterstad、James Tucker 和 Cynthia M. Duncan。

目　录

1 Stata 软件与Stata 的资源

Stata 是用于 Windows、Macintosh 以及 Unix 电脑系统下的一种功能完全的统计软件包。它的特点包括易操作、速度快,还包括一整套预先编好的分析与数据管理功能,同时也允许用户根据需要来创建自己的程序、添加更多的功能。大部分操作既可以通过下拉菜单系统来完成,也可以更直接地通过键入命令来完成。初学者可以在菜单的帮助下学习使用 Stata,任何人在应用自己所不熟悉的程序时都可以由此获得帮助。Stata 的命令有很强的一致性和直观意义,可以使有经验的用户更为高效地工作,这一特点还使得对更复杂或需要多次重复的任务进行编程变得十分容易。如果需要,在应用 Stata 时还可以混用菜单方法和命令方法。它还提供广泛的帮助(**help**)、寻找(**search**)和链接(**link**)功能,轻轻松松便能完成像查询某一命令句法或其他信息这类的事情。

本书先提供一些介绍性信息,然后我们从一段 Stata 应用示范来说明数据分析的"流程",以及怎样使用分析结果。以后的各章将作更为详细的解释。然而,即使没有任何解释,你也可以看到有关命令多么简单明了:打开数据文件 *filename* 的命令就是 **use *filename***,取得概要统计的命令是 **summarize**,取得相关矩阵的命令是 **correlate**,如此等等。并且,也可以通过 **Data** 或 **Statistics** 菜单上的选择来取得同样的结果。

有各种各样的资源来帮助用户学习 Stata,以解决任何层次的困难。这些资源并不只是来自于 Stata 公司,而且也来自于活跃的 Stata 用户群体。本章的一部分内容就是介绍一些最重要的资源:包括 Stata 的热线帮助和打印版的文件,以及在寻求技术帮助时应该给哪里打电话、发传真、写信或发电子邮件。Stata 的网址是 www.stata.com,它提供多种服务,包括软件更新与常见问题解答。此外,还有互联网论坛Statalist Internet,以及专门的索引期刊 *Stata Journal*。

本书体例的说明

本书采用几种不同的印刷体例来标志有关文字的类型意义:

- 凡文中采用粗黑体的英文文字(如 **bold Courier font**)专门表示命令。当给出完整的命令行时,将以一个英文句点作为起始点,这与 Stata 结果窗口显示或输出文件(以 log 为扩展名)中的体例一样。比如:

. **list *year boats men penalty***

- 命令中的变量名(***variable***)或文件名(***file***)均为粗斜体,以强调它们是相机而

定的,并不是命令的固定部分。

- 本书一般行文中涉及变量名(*variable*)或文件名(*file*)时将采用不加粗的斜体,以示它们与一般文字的区别。
- Stata 菜单上的项目将以 **Arial**(一种无饰线的英文字体——译者注)体表示,以破折号表示随后选择。比如,打开一个现有数据文件的菜单选项依次为 **File-Open**,然后找到并点击这一数据集的文件名。注意,一些常规菜单的动作也可以通过 Stata 主菜单工具条中的文字选项来完成:

  File Edit Prefs Data Graphics Statistics User Window Help

  或者点击下面相应的图标钮来完成。比如,选择 **File-Open** 与点击最左侧的图标钮的行动是完全一样的。用户还可以直接键入以下命令来完成同一动作:

```
. use filename
```

- 可以在结果(**Results**)窗口看到的 Stata 输出将采用小号字(如 small Courier font)来表示。小号字可以允许 Stata 的 80 列输出格式能够适合本书的排印宽度。于是,我们显示名为 *penalty* 的变量的概要统计指标的计算结果时,就用以下形式:

```
. summarize penalty

Variable |       Obs        Mean    Std. Dev.       Min        Max
---------+--------------------------------------------------------
 penalty |        10          63    59.59493         11        183
```

这些体例只适用于本书,而不适用于 Stata 本身的程序。Stata 可以显示不同的屏幕字体,但是它在命令中并不使用斜体字。一旦 Stata 的日志(log)文件装载入文字处理软件,或者将结果中的表复制并粘贴到文字处理软件,你应该将其格式改为 Courier 体的 10 号或更小号字,这样才能将各列对应。

需要注意,Stata 对于命令和变量名是区分大小写差别的。所以,**summarize** 就是一个命令,而 Summarize 和 SUMMARIZE 就不是命令。并且,*Penalty* 和 *penalty* 将是两个不同的变量。

一个 Stata 操作的例子

我们先来看一看 Stata 是如何工作的,这一节将介绍如何打开和分析一个以往建立的数据文件,文件名为 *lofoten.dta*。Jentoft 和 Kristofferson (1989)在一篇关于挪威北极圈内 Lofoten 群岛的渔民自我管理的论文中首次发表了这些数据。这个数据中包含 10 次观测(年)和 5 个变量,其中就有 *penalty* 这个变量,它记载了每年渔民违反渔业条例的次数。

如果我们想对这段工作保存一个记录,最好的方法是在工作开始时先打开一个用于输出日志的"log 文件"。log 文件可以存放命令和统计结果表,但是不能存放图形。要建立一个 log 文件,先点击滚轴样的开始 log(**Begin Log**)图标钮,并为这个输出结果的 log 文件设置文件名和文件夹。或者,也可以通过在主菜单工具条上选择 **File-Log-Begin**起始这个文件,还可直接键入以下命令来起始这个文件:

```
. log using monday1
```

在 Stata 中,有多种方式来做同一件事。每一种都有自己的优点,各自适合于不同

场合或不同用户的偏好。

log 文件既可以按一种特殊的 Stata 格式(.smcl)来建立,也可以采用一般文本或 ASCII 格式(.log)。.smcl(即 Stata markup and control language 的缩写)文件格式在使用 Stata 时能很好地浏览和打印。其中还可以包括超链接以方便理解命令或错误提示。一般的 log 文本文件则不能使用这些格式,但是如果用户将来要将这些输出插入其他文档或进行进一步编辑时,就会很方便。用户在选择了所需要的 log 文件类型后,便可以点击 **Save**。在这一节中,我们将建立一个.smcl 格式的 log 文件,将其命名为 *monday1.smcl*。

这里将分析一个现有的 Stata 格式的数据文件 *lofoten.dta*。要打开这个数据,我们仍然有好几种方式:

从主菜单工具条上点击 **File-Open-***lofoten.dta*;

直接点击 -*lofoten.dta*;

键入命令 **use *lofoten***。

在默认 Windows 设置下,Stata 将会在文件夹 C:\data 中寻找数据文件。如果我们想要的文件在别的文件夹中,我们可以在 **use** 命令中定义它的位置:

```
. use c:\books\sws8\chapter01\lofoten
```

或者用命令 **cd**(代表 change directory,即改变子目录)来改变这一阶段的默认文件夹:

```
. cd c:\books\sws8\chapter01\
. use lofoten
```

通常,取得文件的最简单方法是选择 **File-Open**,然后按常规方式浏览该文件夹加以选择。

如果想要取得现在已经在内存中的数据的简要描述,键入:

```
. describe

Contains data from C:\data\lofoten.dta
  obs:            10                          Jentoft & Kristoffersen '89
 vars:             5                          30 Jun 2005 10:36
 size:           130 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %9.0g                  Year
boats           int    %9.0g                  Number of fishing boats
men             int    %9.0g                  Number of fishermen
penalty         int    %9.0g                  Number of penalties
decade          byte   %9.0g       decade     Early 1970s or early 1980s
-------------------------------------------------------------------------------
Sorted by:  decade  year
```

许多 Stata 命令都可以简化为它们的前几个字母。比如,我们可以将 **describe** 命令简化为仅有一个字母 **d**。如果要使用菜单,那么选择 **Data-Describe data-Describe variables in memory-OK** 也能得到同样的输出表格。

这一数据只有 10 个观测和 5 个变量,所以键入 **list** 就能列出相应内容(或者就键入小写字母 **l** 也行;或者选择 **Data-Describe data-List data-OK** 也行):

```
. list

     +------------------------------------------+
     | year   boats    men   penalty   decade   |
     |------------------------------------------|
  1. | 1971    1809   5281        71    1970s   |
  2. | 1972    2017   6304       152    1970s   |
  3. | 1973    2068   6794       183    1970s   |
  4. | 1974    1693   5227        39    1970s   |
  5. | 1975    1441   4077        36    1970s   |
     |------------------------------------------|
  6. | 1981    1540   4033        11    1980s   |
  7. | 1982    1689   4267        15    1980s   |
  8. | 1983    1842   4430        34    1980s   |
  9. | 1984    1847   4622        74    1980s   |
 10. | 1985    1365   3514        15    1980s   |
     +------------------------------------------+
```

我们从平均值(Mean)、标准差(Std. Dev.)、最小值(Min)以及最大值(Max)入手来进行分析(直接键入 **summarize** 或 **su**;或者选择 **Statistics-Summaries, tables, & tests-Summary statistics-Summary statistics-OK**):

```
. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        year |        10        1978    5.477226       1971       1985
       boats |        10      1731.1    232.1328       1365       2068
         men |        10      4854.9    1045.577       3514       6794
     penalty |        10          63    59.59493         11        183
      decade |        10          .5    .5270463          0          1
```

如果需要将这部分结果打印出来,那么首先点击结果(**Results**)窗口将其移到前面来,或者是点击图标钮(即 **Bring Results Window to Front**)也行,然后点击图标钮(即 **Print**)。

如果想复制一个表、一些命令,或结果窗口的其他信息到文字处理软件中,首先要将结果窗口移到前面来(点击这个窗口或图标钮)。然后用光标选择想要的那些结果,点击光标右键,再选择光标菜单上的 **Copy Text**。最后,转到你的文字处理软件中,在适当插入点点击光标右键、然后点击 **Paste**。或者,点击该文字处理器工具条上的"剪粘板(**clipboard**)"图标钮也行。

那么在这个数据包括的20年中违反渔业条例的处罚次数是否有所变化呢?我们可以对每10年(*decade*)的处罚(*penalty*)做概要统计,结果显示出1970年代有更多的处罚:

```
. tabulate decade, sum(penalty)

Early 1970s |
   or early |  Summary of Number of penalties
      1980s |        Mean   Std. Dev.       Freq.
------------+------------------------------------
      1970s |        96.2    67.41439           5
      1980s |        29.8   26.281172           5
------------+------------------------------------
      Total |          63   59.594929          10
```

同一个表也可以通过菜单选择来取得:**Statistics-Summaries, tables, & tests-Tables-One/two-way table of summary statistics**,然后将 *decade* 作为变量1(**variable 1**)填入,而将 *penalty* 作为概要统计变量(**variable to be summarized**)。尽管使用菜单选择通常都很简单明了,但是你能看

到在描述它们时却比使用简单文字命令更复杂。因此,后面我们将主要使用命令,只在少许场合提及菜单选用。对于菜单的探究、搞清其如何使用才能完成同样的任务,将留给读者自己来完成。出于同样的原因,Stata 参考手册(*Stata reference manuals*)也是采取了以命令为基础的方式。

也许,处罚次数的减少是因为在 **1980** 年代打鱼的人变少了。我们发现,处罚次数与同期渔船数(*boats*)和渔民人数(*men*)之间存在着高度相关($r>0.8$):

```
. correlate boats men penalty
(obs=10)

             |    boats      men  penalty
-------------+---------------------------
       boats |   1.0000
         men |   0.8748   1.0000
     penalty |   0.8259   0.9312   1.0000
```

图形可以更清楚地反映它们之间的关系。图 1.1 按年(*year*)画出了 *men* 与 *penalty* 的标绘图,命令为 **graph twoway connected**。在这个例子中,我们先要求将按年对 *men* 作双变量连线(connected-line)标绘图,定义了左侧 y 轴,**yaxis(1)**。在分隔符 || 以后,我们又要求按年对 *penalty* 作连线图,这次定义右侧 y 轴,**yaxis(2)**。结果图形表明,渔民人数与处罚次数在时间上有对应关系。

```
. graph twoway connected men year, yaxis(1)
      || connected penalty year, yaxis(2)
```

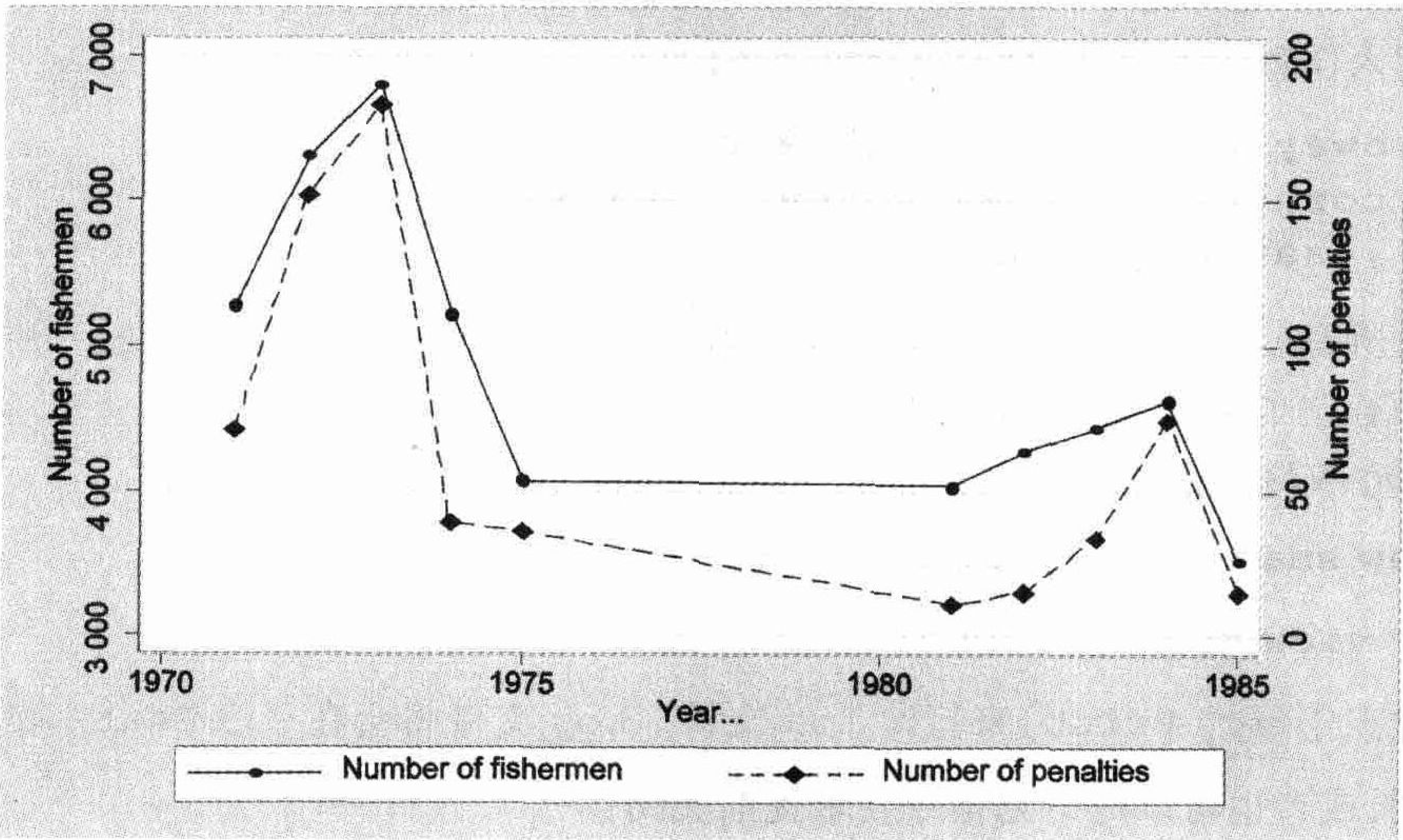


图 1.1

由于 1976 年至 1980 年的数据有缺失,图 1.1 显示中将 1975 年直接连接到 1981 年。有时出于种种原因,我们不太愿意这样做。作为备择方案,我们既可以去找到缺失数据,也可以采用稍微复杂一点的命令在这段时期留一个不连接的缺口。

要打印出这个图,点击 **Graph** 窗口或点击 (将图形窗口移前),然后点击打印图标钮 。

要将此图直接复制到文字处理器或其他文件中,先将图形窗口移前,右键点击这一图形,然后选择 **Copy**。再转到你的文字处理器窗口,定位插入点后,选择一种适当的粘贴方式,比如 **Edit-Paste** 或 **Edit-Paste Special(Metafile)**、或直接点击"剪粘板(**clipboard**)"图标(不同的文字处理器有不同的处理方式)。

如果需要将此图存起来将来再用，可以右键点击并选择 **Save**，或选择主菜单上的 **File-Save Graph**。在 **Save As Type** 子菜单可以选择存为几种不同的文件格式。在 **Windows** 系统中，这些选项包括：

Stata graph (* .gph)(一种“活”的图形，包括足够的信息供 Stata 来编辑)

As-is graph (* .gph)(一种更压缩的 Stata 图形格式)

Windows Metafile (* .wmf)

Enhanced Metafile (* .emf)

Portable Network Graphics (* .png)

TIFF (* .tif)

PostScript (* .ps)

Encapsulated PostScript with TIFF preview (* .eps)

Encapsulated PostScript (* .eps)

不管我们需要哪种图形格式，都值得同时再存一份这一图形的“活”的格式，即 .gph 格式。这种 .gph 格式在将来还可以用 **graph use** 或 **graph combine** 命令来重新打开、合并、重新着色或重新设置格式(参见第 3 章)。

除了使用菜单以外，也可以在任何 **graph** 图形命令之后加上 **saving(*filename*)** 选项来存为图形文件。比如，要把图形存为文件名为 *figure1.gph* 的文件，就在原制图命令后加入另一个分隔符、一个英语逗号以及 **saving(*figure1*)** 即可。第 3 章将会对 **graph** 命令的原理做更多的解释。现在这个完整的命令包括以下内容(在 Stata 命令窗口可以键入任意多的空格，只要没有硬回车即可)：

```
. graph twoway connected men year, yaxis(1)
      || connected penalty year, yaxis(2)
      || , saving(figure1)
```

通过以上所有分析，log 文件 *monday1.smcl* 中已经存放了我们的结果。用好几种方法都能检查这个文件，看看我们曾经做过什么：

File-Log-View-OK

-View snapshot of log file-OK

键入命令 **view monday1 .smcl**

我们可以通过点击 (**Print**)来打印 log 文件。Log 文件将会在一段 Stata 操作完成后自行关闭，或者也可以用下列指令要求提前关闭：

File-Log-Close

-Close log file-OK

键入命令 **log close**

一旦关闭，文件 *monday1.smcl* 就可以再通过随后的 Stata 操作的 **File-View** 再打开。为了使输出文件能更容易地被其他的文字处理器打开，可以键入以下命令将 log 文件从 .smcl 格式(Stata 格式)转换为 .log 格式(标准 ASCII 文本格式)：

```
. translate monday1.smcl monday1.log
```

或者，一开始就建立 .log 格式文件而不用 .smcl 格式。

Stata 的文件管理与帮助(Help)文件

Stata 第 9 版的整套文件包括 15 卷,共计超过 6 000 页:一本较薄的《初学手册》(比如,*Getting Started with Stata for Windows*),一本更广泛的《用户指南》(*User's Guide*),三卷本的《基础参考手册》(*Base Reference Manual*),还有分别对数据管理、制图、纵贯和面板数据、矩阵编程(Mata)、多元统计、编程、调查数据、存活分析和流行病学梯度表,以及时间序列分析的参考手册。《初学手册》只是帮助用户做最基本的安装、视窗管理、数据输入、打印等方面的工作。《用户指南》是对一般问题的更广泛的讨论,包括资源与问题解决。新用户尤其要注意的是《用户指南》中的一节:"所有人都应该知道的命令(Commands everyone should know)"。《基础参考手册》按字母排列出了所有 Stata 命令。每一条命令都包括了完整的命令句法、所有可用选项的描述、例子、有关公式和基本原理的技术说明,以及其他参考文献。数据管理、制图、面板数据等在一般参考文献中已经涉及了,但是更复杂的题目是在它们自己的专题手册中才提供更具体的处理方法以及例子。还有一本《快速参考与索引》(*Quick Reference and Index*)提供了全部文件齐全的清单。

当我们在操作 Stata 时,更简单的是取得在线帮助而不是去查询这些手册。从主菜单工具条选择 **Help** 时将会拉下更多选择的菜单,包括对特定命令的帮助、一般问题、在线更新、Stata 期刊,以及连接 Stata 的网址(www.stata.com)。此外,我们也可以将浏览器窗口(Viewer)移到前面(或点击),并使用其检索(**Search**)和内容(**Contents**)的功能来寻找信息。我们还可以使用 **help** 命令。比如,键入 **help correlate** 命令将使有关帮助信息显示在浏览器窗口。与参考手册一样,屏幕帮助也提供命令句法说明以及完整的选项清单。它还包括了一些例子,但常常不太具体,而且不提供手册中那些技术讨论。但是,浏览器帮助相比手册也有一些优点。它能够在 Stata 互联网址的文件中搜寻关键词。超级文本链接可以使你直接找到有关条目。屏幕帮助还包括一些最近更新的资料,或者你还可以从 Stata 网址或其他用户网址下载一些"非官方"的 Stata 程序。

搜寻信息

选择 **Help-Search-Search documentation and FAQs** 提供一个直接方式来搜寻 Stata 文件资料中的信息或网址上的常见问题解答(FAQs,即 frequently asked questions)和其他页面。相应的 Stata 命令是:

```
. search keywords
```

与命令 **search** 相关的选项允许我们规定搜寻范围,比如,搜寻 Stata 文件和常见问题解答,或者搜寻网上资源,包括其期刊(*Stata Journal*),或者同时搜寻这两个资源。比如:

```
. search median regression
```

这个命令将搜寻文件和常见问题中与关键词"median(中位数)"和"regression(回归)"有关的信息。如果还想将搜寻范围从文件资料和常见问题进一步扩大到 Stata 的更多网上资源,就键入

```
. search median regression, all
```

浏览器窗口的搜寻结果将包括可点击的超链接到更多信息或原著引用。

对命令 **search** 的一种特殊使用在某些场合下会提供更多信息，比如，当我们的命令没有被成功执行因而导致得到的是含义不明的 Stata 错误提示码。比如，键入一个单词的命令 **table** 就会得到错误提示或"返回码(return code)"r(100)：

```
. table
varlist required
r(100);
```

这是因为命令 **table** 显然是需要附上变量表的。但是，错误提示的意义往往并不太清楚。如果想知道返回码 **r(100)** 到底是什么意思，可键入：

```
. search rc 100
Keyword search

        Keywords:  rc 100
          Search:  (1) Official help files, FAQs, Examples, SJs, and STBs

Search of official help files, FAQs, Examples, SJs, and STBs

[P]     error . . . . . . . . . . . . . . . . . . . . . . . . Return code 100
        varlist required;
        = exp required;
        using required;
        by() option required;
        Certain commands require a varlist or another element of the
        language.  The message specifies the required item that was
        missing from the command you gave.  See the command's syntax
        diagram.  For example, merge requires using be specified; perhaps,
        you meant to type append.  Or, ranksum requires a by() option;
        see [R] signrank.

(end of search)
```

键入 **help search** 可以提供关于这个命令的更多信息。

Stata 公司

要搜寻有关定购、许可证和更新方面的信息，你可以通过下列电子邮箱与 Stata 公司联系：

stata@ stata.com

或者访问他们的网站：

http://www.stata.com

Stata 网站有丰富的用户支持信息，并且还提供其他资源的链接。Stata 出版社(Stata Press)还有其自己单独的网站，提供关于 Stata 出版物的信息，包括例题所用的数据。网址为：

http://www.stata-press.com

这两个网站都很值得进行探究。

Stata 公司的邮寄地址是：

Stata Corporation

4905 Lakeway Drive

```
College Station, TX 77845 USA
```

电话号码也包括很好记的 **800** 号码。

```
telephone:1-800-STATAPC  U.S.
          (1-800-782-8272)
          1-800-248-8272  Canada
          1-979-696-4600  International
fax:      1-979-696-4601
```

对于有许可证的 Stata 用户,多数版本的软件在线升级是免费的。这就为用户当前版本取得最新改进、错误修复等提供了便捷的途径。如果想查一查自己的 **Stata** 是否需要更新了,就键入以下命令来启动自动在线升级进程:

```
. update query
```

要寻求技术帮助,用户可以通过电子邮件询问,在标题行中要写明你的Stata序列号:

tech_support@ stata.com

不过,在进行电话联系或写信寻求技术援助以前,用户也许应该先链接到 www.stata.com看看你的问题是否已经在常见问题中解答过了。这个网址还提供产品、定购以及帮助信息;国际语言说明;分类新闻与公告。更多的是提供用户支持,包括以下服务:

FAQS——常见问题解答。如果用户有什么困扰,并且在手册中又找不到答案,那么就可以查查这里。也许它就是一个常见问题。这里的问答涉及面很宽,既有很基础的问题,像"如何将其他软件文件转换为 Stata 格式的数据文件?";也有更技术化的问题,比如,"如何在完全最大似然估计中使用 **heckman** 命令来强制 $\rho = 0$?"

UPDATES——更新升级。对于有许可证的 Stata 用户,经常性的较小更新或错误修复,可以免费下载。

OTHER RESOURCES——其他资源。链接和信息中包括在线 Stata 教学(网上课程,NetCourses);Stata 期刊的增进;Stata 用户进行讨论的独立名单服务器(Statalist);销售有关 Stata 的图书和其他最新统计参考资料的书店;下载与 Stata 图书相关的数据与程序;通向其他统计网站的链接,其中也包括 Stata 的竞争者。

下面一节来描述一些最重要的用户支持资源。

Statalist

Statalist 提供了一个极有价值的 Stata 活跃用户之间联系的在线论坛。它独立于 Stata 公司,尽管 Stata 的程序员们对其进行监察,并且也经常参与讨论。要订阅 Statalist,就给以下电子邮箱发个邮件:

majordomo@ hsphsun2.harvard.edu

邮件内容只需要写以下一段话即可:

subscribe statalist

于是名单处理器就会承认接到你的来信并附上如何使用这个名单的说明,包括如何将你自己的消息张贴到论坛上去。任何发送到下列电子邮箱的消息都会寄到当前所有订阅者处:

statalist@ hsphsun2.harvard.edu

千万不要试图通过直接给 Statalist 地址发订阅或取消订阅的通知。这并不能达

到你的目的,但是却会将你的错误分发给成百上千的订阅者。要想从名单上取消订阅,请同样写给你订阅时用过的 majordomo 邮箱:

```
majordomo@ hsphsun2.harvard.edu
```

但是内容只写以下一段话:

```
unsubscribe statalist
```

或者是同一意思的另一表达:

```
signoff statalist
```

如果你计划外出旅行一段时间,取消订阅将保证你的邮箱不致被 Statalist 的消息塞满。你总是可以重新订阅。

要搜寻 Statalist 档案,可以链接

```
http://www.stata.com/statalist/archive/
```

Statalist 的材料包括索取程序、求解方法、有关建议,以及回答和一般讨论。与 Stata 期刊(下面讨论)一道,Statalist 在扩展 Stata 本身能力以及认真的 Stata 用户的能力方面发挥了主要作用。

专门期刊 Stata Journal

从 1991 年至 2001 年,称为 *Stata Technical Bulletin*(简称 STB)的双月刊服务于发布新的命令和 Stata 更新,其中既有用户撰写的,也有正式渠道发布的。STB 上的文章累积起来,每年都出版一本书,称为 Stata Technical Bulletin Reprints,这些书可以从 Stata 公司直接订购。

随着网络的发展,用户之间通过 Statalist 这种载体的即时交流成为可能。程序文件能从遥远的资源地轻易下载。双月刊印的期刊和磁盘对于用户交流或发布更新与用户撰写的程序而言,都已经不再是最好的途径了。为了适应变化了的世界,STB 也必须有新的发展。

于是,*Stata Journal*(Stata 期刊)开始发行,以迎接挑战、满足 Stata 日益扩大的用户群。像以前的 STB 一样,*Stata Journal* 仍包括用户描述研制新命令的文章,也包括 Stata 公司雇员编制的非正式命令。但是,发布新命令并不是它的首要关注。*Stata Journal* 还包括带索引的统计学注释文章、书评,以及一些有趣的栏目,比如,由 Nicholas J. Cox 主持的"话说 Stata"(Speaking Stata)讨论如何更有效率地使用 Stata 编程语言。*Stata Journal* 既给初学者服务,也给老用户服务。比如,这里是最近一期的目录:

Stata Journal 是每季度发行,可以通过访问 www.stata.com 直接向 Stata 公司购买订阅。

应用 Stata 的图书

除了 Stata 自己的参考手册以外,描述 Stata 或应用 Stata 来示范分析技术的书目越来越多。这些书中包括一般性介绍;学科应用,如社会科学、生物统计或经济计量;以及有关调查分析、实验数据、分类因变量以及其他题目的专门著述。Stata 网页上的书店提供最新的书目清单,并且附有内容描述:

http://www.stata.com/bookstore/

这个网上书店提供了一个了解和订购不同出版商发行的 Stata 相关图书的好地方。

2 数据管理

数据分析的第一步就涉及将原始数据改造为 Stata 可用的格式。我们可以将一个新数据通过以下几种方式载入 Stata:从键盘上将数据输入;读取原始数据的 ASCII 格式文件;将电子表格数据粘贴到数据编辑窗口(Editor)中;应用第三方数据转换程序将其他电子表格、数据库或统计程序创建的系统数据集直接转换过来。一旦 Stata 有了内存数据,我们就可以在 Stata 中将其存为 Stata 格式,以利将来方便地取用和更新。

数据管理包括最初建立数据集、编辑和校正错误以及内部建档,比如,加上变量标签和变量值标签。它也包括许多其他项目进行中所需要的工作,比如,加入新的观测案例或新的变量;重新组织数据、简化数据、从数据中抽样;分割、合并或拆分数据;改变变量类型;通过代数或逻辑计算建立新的变量。当数据管理任务很复杂或需要重复进行时,Stata 用户可以编写自己的程序来自动完成这些工作。尽管 Stata 是因其分析功能而著名,其实它同时也具有广泛的数据管理功能。本章将介绍其中一些基本功能。

《用户指南》(*User's Guide*)提供了多种数据输入方法的总览,并建议了决策采用哪一种方法的八条规则。本章所讲的输入、编辑和许多其他操作都可以通过数据菜单(**Data**)来完成。数据菜单的下属标题指点了总的任务分类:

Describe data	描述数据
Data editor	数据编辑器
Data browser (read-only editor)	数据浏览器(只读编辑器)
Create or change variables	创建或修改变量
Sort	排序
Combine datasets	合并数据集
Labels	标签
Notes	说明
Variable utilities	变量用途
Matrices	矩阵
Other utilities	其他用途

命令示范

. append using *olddata*

读入以前所存的数据集 *olddata.dta*,然后将其所有观测加到当前内存中的数据中去。随后键入 **save *newdata*, replace** 就能将这一合并数据集存为新数据文件 *newdata.dta*。

```
. browse
```

打开表格化的数据浏览器(**Browser**)来查看数据。浏览器看起来很像数据编辑器(**Data Editor**),但是它没有编辑功能,所以也就没有一不小心改变数据的风险。这一操作的替换方法是点击图标。

```
. browse boats men if year > 1980
```

要求打开数据浏览器时只显示 year 变量取值大于 1980 的那些观测案例的 *boats* 和 *men* 变量值。这个命令示范了 **if**(如果)的选择功能,它还可以用于许多 Stata 命令的选择操作。

```
. compress
```

自动地将所有变量转换为其最有效率的存储类型以节省内存和磁盘空间。随后键入命令 **save *filename*, replace** 将使这些改变永久化。

```
. drawnorm z1 z2 z3, n(5000)
```

创建一个人工数据集,包含从独立的标准正态分布中抽取的 5 000 个观测案例和 3 个随机变量 *z1*,*z2*,*z3*。还可以通过选项命令定义其他的平均数、标准差、相关矩阵或协方差矩阵。

```
. edit
```

打开表格化数据编辑器,以便进行数据输入或编辑。这一操作的替换方法则是选择 **Window-Data Editor** 或直接点击图标。

```
. edit boats year men
```

打开数据编辑器时,只显示 *boats*、*year*、*men* 等 3 个变量(而且就按这一顺序),以便加以编辑。

```
. encode stringvar, gen(numvar)
```

根据字符型(非数量型)变量 *stringvar*,新建一个有标签的数量型变量,名为 *numvar*。

```
. format rainfall %8.2f
```

为数量型变量 *rainfall* 建立一种固定化(**f**)的显示格式,即 8 列宽,小数点后显示 2 位数。

```
. generate newvar = (x + y)/100
```

建立一个名为 *newvar* 的新变量,其值等于 *x* 加上 *y* 后再除以 100。

```
. generate newvar = uniform()
```

建立一个名为 *newvar* 的新变量,其值从一个随机均匀分布的 0 到接近 1 区间中取样,记为[0,1)。

```
. infile x y z using data.raw
```

读入一个名为 *data.raw* 的 ASCII 文件,其中包含 3 个变量 *x*,*y*,*z*。这些变量值由一个或多个空格分隔开,或者是由制表符、回车符、换行符分隔,或者是由英文逗号分隔。如果是由空格做分隔符的,那么缺失值是由英文句点代表,而不是由空格代表。要是采用逗

号分隔符,缺失值则由一个句点或两个连续的逗号来代表。Stata 还提供了更多的缺失值处理功能,我们将在后面加以讨论。有一些其他命令更适合于读取制表分隔符、逗号分隔符或固定列格式的原始数据。键入 **help infiling** 可以取得更多的信息。

. list

按默认或"表格"格式列出数据。如果数据中有许多变量,表格格式就很难审阅,那么 **list, display** 可输出更好的结果。参见 **help list** 中其他有关控制数据表格式的选项。

list *x y z* in 5/20

按照当前的数据顺序,列出第 5 至第 20 个观测案例的 *x*,*y*,*z* 三个变量值的清单。这种 **in** 方式的选择功能在大多数 Stata 命令中也能同样应用。

. merge *id* using *olddata*

读入以前所存的数据集*olddata.dta* ,然后将*olddata* 中的观测与内存中的具有同样*id* 值的观测加以匹配。在此项操作之前,*olddata* 中的观测案例(称为"使用(using)"数据)和当前在内存中数据(称为"主(master)"数据)都必须已经按 *id* 值排好顺序了。

. replace *oldvar* = 100 * *oldvar*

将变量 *oldvar* 的原值扩大100 倍后再取代原值。

. sample 10

将内存中所有观测案例只随机选取 10% 样本留下,其他观测案例全数删除。除了可以按某一百分比抽取样本外,我们还可以选择某一数量的案例。比如,**sample 55, count** 就能删除其他观测案例、仅保留 55 个观测案例的随机样本。

. save *newfile*

将当前内存中的数据存为一个新数据文件 *newfile.dta*。如果 *newfile.dta* 已经存在,而你又想覆盖以前版本,那么键入 **save *newfile*, replace**。替换方法是,在菜单上选择 **File-Save** 或 **File-Save As**。如果要把 *newfile.dta* 存为 Stata 第7版格式,可键入 **saveold *newfile***。

. set memory 24m

(只适用于 Windows 或 Unix 系统。)为 Stata 数据分配 24 兆字节内存。分配数量可以更大,也可以更小。当需要量超过了物理内存时,就会采用虚拟内存(硬盘空间)。在使用 **set memory** 命令前,键入 **clear** 以便从内存中清除当前数据。

. sort *x*

将数据按 *x* 值从最小到最大依次排序。那些 *x* 值缺失的观测案例将排在最后,因为 Stata 将缺失值当作非常大的值来处理。键入 **help gsort** 可以了解完成更一般化的排序任务的命令,比如,可以选择按升序排还是降序排,也可以专门将缺失值排到最前面来。

. tabulate *x* if *y* > 65

只对那些 *y* 值大于 65 的观测案例输出 *x* 的频数表。这里 **if** 的选择功能与在大多数其他 Stata 命令中一样。

```
. use oldfile
```

找到磁盘上以前所存的 Stata 格式数据 *oldfile.dta*,将其置于内存中。如果当前有其他数据在内存中,并且你并不想保存就放弃,那么键入 **use *oldfile*, clear**。用替换方法,选择 **File-Open** 或点击 也可以完成同样任务。

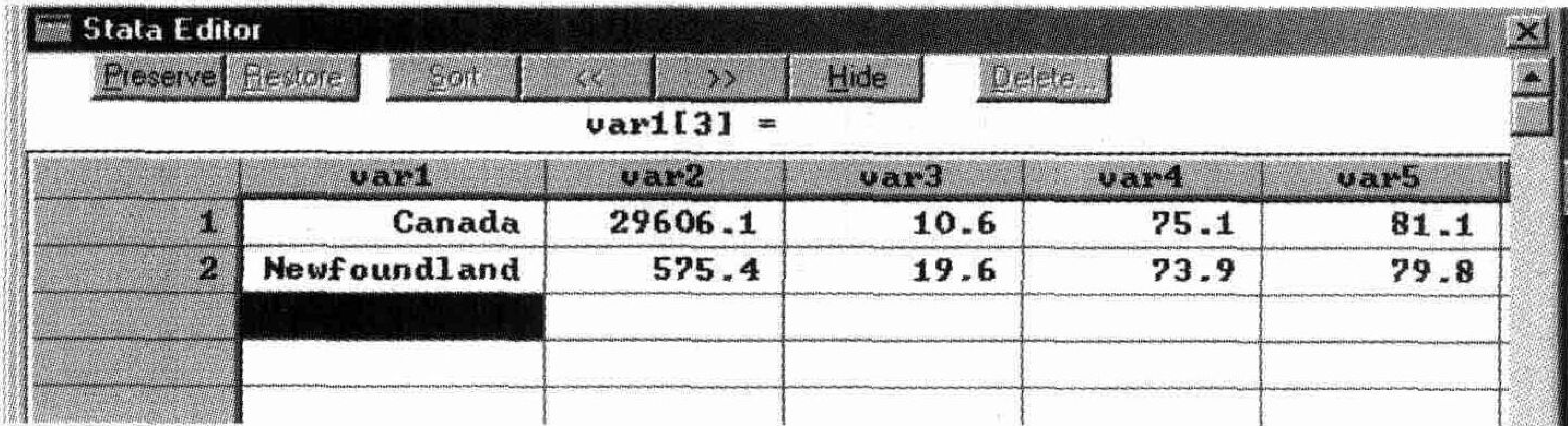
创建一个新数据

以前所存的 Stata 格式数据可以取出并置入内存中,或者采用键入命令 **use *filename*** 的方法,或者采用菜单选择方法。本节将描述创建一个全新的 Stata 格式数据的基本方法,借助表 2.1 中所列 1995 年加拿大各省及领土区域数据来加以示范(取自联邦、省级及领土区域人口健康咨询委员会,1996 年。加拿大的最新领土 Nunavut 没有列在其中,因为它在 1999 年以前只是西北领土(Northwest Territories)的一部分)。

表 2.1 加拿大及各省数据

Place	1995 Pop. (1000's)	Unemployment Rate(percent)	Male Life Expectancy	Female Life Expectancy
Canada	29 606.1	10.6	75.1	81.1
Newfound land	575.4	19.6	73.9	79.8
Prince Edward Island	136.1	19.1	74.8	81.3
Nova Scotia	937.8	13.9	74.2	80.4
New Brunswick	760.1	13.8	74.8	80.6
Quebec	7 334.2	13.2	74.5	81.2
Qntario	11 100.3	9.3	75.5	81.1
Manitoba	1 137.5	8.5	75.0	80.8
Saskatchewan	1 015.6	7.0	75.2	81.8
Alberta	2 747.0	8.4	75.5	81.4
British Columbia	3 766.0	9.8	75.8	81.4
Yukon	30.1	—	71.3	80.4
Northwest Territories	65.8	—	70.2	78.0

要创建表 2.1 这样的数据,最简单的方法是通过 Stata 表格化的数据编辑器(**Data Editor**),只要点击图标 就可以调用,也可在顶部菜单条中选择 **Window-Data Editor**,或者直接键入命令 **edit**。Stata 会自动生成 *var1*、*var2* 等变量列,用户就可以开始为每个变量键入数据。于是,*var1* 包含地名(如 Canada、Newfoundland 等),*var2* 是人口数,以此类推。



我们可以定义更有意义的变量名,只要双击相应列的标题(如 *var1*),然后在所启动的对话框中键入新变量名即可。尽管变量名最多允许 32 个字符,但是最好保持在 8

个或更少字符。我们还可创建包含简要描述的变量标签。比如,*var2*（人口）可以重新命名为 *pop*,并建立相应的变量标签为“Population in 1000s, 1995”。

变量的重新命名和制作标签也可以在数据编辑器之外通过 **rename** 和 **label variable** 命令来完成,如:

```
. rename var2 pop
. label variable pop "Population in 1000s, 1995"
```

那些空着的单元格,比如,Yukon 和 Northwest Territories 的失业率将会自动生成 Stata 的系统(默认)缺失值码,即一个点。我们在任何时候都可以关闭数据编辑器,然后将数据存入磁盘。点击 ▦ 或选择 **Window-Data Editor** 就可以重新返回数据编辑器。

如果为某一变量输入的第一个值是一个数字,比如,对人口、失业率和预期寿命这些变量,那么 Stata 便会认为这一列是“数值变量”,从此以后只允许输入数值。数值也可以带正负号,也可以包括小数点,也可以采用科学记数法。比如,将加拿大人口表示为 2.960 61e +7,它表示 2.96061×10^7,即大约 2 960 万人。输入数值时**不能包含任何逗号**,比如,29 606 100。要是我们偶然在某列第一次输入的数值中加入了逗号,那么 Stata 将认为本列是“字符串变量”(参见下一段),而不将其作为数值对待。

如果为某一变量第一次输入的是非数值字符,比如,像地名的输入(或者输入了带逗号的“1 000”),那么 Stata 会判断此列是字符串变量。字符串变量的值几乎可以是任何字母、数字、符号或空格的组合,在 Intercooled 版本或 Small Stata 版本中,长度限制为 80 个字符,在 Stata/SE 版本中允长可达 244 个字符。于是,我们就可以在其中存放名称、引用语或其他描述性信息。字符串变量的值可以列表和计数,但是不能计算平均数、相关系数或大多数其他统计量。在数据编辑器或数据浏览器中,字符串变量值显示为红色,所以我们很容易区分这两类变量。

按以上方式将表 2.1 的信息输入完毕后,我们便关闭数据编辑器并存储我们的数据,将文件命名为 *canada0.dta*:

```
. save canada0
```

Stata 将自动加上文件扩展名.dta,除非我们要它不这样做。如果我们以前就存过同名文件的早期版本,那么想要以新版本覆盖原有版本,可以键入:

```
. save, replace
```

这时,我们的新数据看起来就像这样:

```
. describe

Contains data from C:\data\canada0.dta
  obs:            13
 vars:             5                          3 Jul 2005 10:30
 size:           533 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
var1            str21  %21s
pop             float  %9.0g                  Population in 1000s, 1995
var3            float  %9.0g
var4            float  %9.0g
var5            float  %9.0g
-------------------------------------------------------------------------------
Sorted by:
```

```
. list

     +-----------------------------------------------------------------+
     |                  var1       pop   var3   var4   var5 |
     |-----------------------------------------------------------------|
  1. |                Canada   29606.1   10.6   75.1   81.1 |
  2. |          Newfoundland     575.4   19.6   73.9   79.8 |
  3. |  Prince Edward Island     136.1   19.1   74.8   81.3 |
  4. |           Nova Scotia     937.8   13.9   74.2   80.4 |
  5. |         New Brunswick     760.1   13.8   74.8   80.6 |
     |-----------------------------------------------------------------|
  6. |                Quebec    7334.2   13.2   74.5   81.2 |
  7. |               Ontario   11100.3    9.3   75.5   81.1 |
  8. |              Manitoba    1137.5    8.5     75   80.8 |
  9. |          Saskatchewan    1015.6      7   75.2   81.8 |
 10. |               Alberta      2747    8.4   75.5   81.4 |
     |-----------------------------------------------------------------|
 11. |      British Columbia      3766    9.8   75.8   81.4 |
 12. |                 Yukon      30.1      .   71.3   80.4 |
 13. | Northwest Territories      65.8      .   70.2     78 |
     +-----------------------------------------------------------------+

. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        var1 |         0
         pop |        13    4554.769    8214.304      30.1    29606.1
        var3 |        11    12.10909    4.250048         7       19.6
        var4 |        13    74.29231    1.673052      70.2       75.8
        var5 |        13    80.71539    .9754027        78       81.8
```

检查这些输出表可以使我们查看一下是否输入数据有错,需要加以改正。比如,**summarize** 输出表提供了好几项校对时很有用的信息,包括非缺失观测的频数(对于字符串变量这一频数总是为 0)、各变量的最小值和最大值。此时,这些概要统计还没有实际意义,因为这一数据中有一个观测案例(加拿大)其实代表了所有其他 12 个省和领土区域的总和。

下一步就是使我们数据更加自言其明。变量名可以改得更加有意义,比如:

```
. rename var1 place
. rename var3 unemp
. rename var4 mlife
. rename var5 flife
```

Stata 还允许我们为数据加上好几种类型的标签。命令 **label data** 用于对整个数据的描述。比如:

```
. label data "Canadian dataset 0"
```

命令 **label variable** 只描述单个变量。比如:

```
. label variable place "Place name"
. label variable unemp "%15+ population unemployed, 1995"
. label variable mlife "Male life expectancy years"
. label variable flife "Female life expectancy years"
```

通过制作数据和变量的标签,这个数据就取得了更强的自我解释性:

```
. describe
```

```
Contains data from C:\data\canada0.dta
  obs:            13                          Canadian dataset 0
 vars:             5                          3 Jul 2005 10:45
 size:           533 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
place           str21  %21s                   Place name
pop             float  %9.0g                  Population in 1000s, 1995
unemp           float  %9.0g                  % 15+ population unemployed,
                                                1995
mlife           float  %9.0g                  Male life expectancy years
flife           float  %9.0g                  Female life expectancy years
-------------------------------------------------------------------------------
Sorted by:
```

当标签制作完成后，我们应当用 **File-Save** 或通过键入命令来存放这个数据：

```
. save, replace
```

以后我们就可以通过点击随时调用这些数据，或者用 **File-Open** 或键入：

```
. use c:\data\canada0
(Canadian dataset 0)
```

然后我们可以做一些新的分析。比如，我们可能注意到，男性和女性的预期寿命之间存在正相关，而又与失业率存在负相关。并且在预期寿命与失业率的相关上，男性显得更强一些。

```
. correlate unemp mlife flife
(obs=11)

        |    unemp    mlife    flife
--------+---------------------------
   unemp|   1.0000
   mlife|  -0.7440   1.0000
   flife|  -0.6173   0.7631   1.0000
```

数据中各条观测案例的顺序可以通过命令 **sort** 来改变。比如，要想按人口数由小到大来排序，键入：

```
. sort pop
```

字符串变量无法定量排序，所以将按字母顺序来进行排序。键入 ***sort place*** 就会改变观测的顺序，将 Alberta 排在第一，将 British Columbia 排在第二，如此等等。

我们能够用命令 **order** 来控制变量的顺序。比如，我们可以将失业率这个变量排在第二而将人口数排在最后：

```
. order place unemp mlife flife pop
```

数据编辑器也有一些按钮来执行这些功能。在用光标选择了某一列后，点击 **Sort** 按钮就会按此列排序。而 **< <and > >**按钮则可以将当前选择变量分别移动到变量表的左右两端位置。正如其他编辑工作一样，这些改变只有在经过存盘之后才会成为永久性的。

数据编辑器的 **Hide** 按钮并不重置数据，但是可以使某一列从数据表中暂时不显示。在我们需要输入很多变量但想使地名（或其他案例识别码）总能看见的场合下，这种功能就提供了很大方便，将这些变量置于我们正要输入的变量旁边。

我们还可以事先限制数据编辑器只能对某些变量工作，并按特定的变量顺序，或者只选择某个变量值范围。比如：

```
. edit place mlife flife
```

或者

```
. edit place unemp if pop > 100
```

后一个例子应用了 **if** 选择条件，这就是下一节中将要介绍的一个重要的工具。

定义数据的子集：in 和 if 选择条件

许多 Stata 命令都可以限制为对数据的一个子集来执行，这就需要在命令中加上 **in** 或 **if** 选择条件（这种条件在许多菜单选择里也是提供的，注意寻找 **if/in** 或**by/if/in** 按钮）。选项 **in** 指定了命令应用的观测案例编号。比如，**list in 5** 告诉 Stata只列出第 5 条观测案例。要想列出第 1 至第 20 条观测案例，可以键入命令：

```
. list in 1/20
```

字母 **l** 用来标志最后一个案例，而 **-4** 则表示从最后开始倒数的第 4 个。于是，我们可以列出加拿大人口最多的 4 个地方（其中也包括了加拿大自己），命令如下：

```
. sort pop
. list place pop in -4/l
```

这里特别需要注意，在 **1**（数字 1 或第 1 个案例）与 **l**（小写字母"el"或最后一个案例）的不同，因为它们在印刷效果上较难区别。选择条件 **in** 在大多数其他分析或数据编辑命令中也能应用。另外，应用这个选择条件时，我们应当保证数据已经事先排好序了。

选择条件 **if** 也有广泛的应用价值，但是它是按特定的变量值来进行选择。正如所见，*canada0.dta* 的观测案例不仅包括了加拿大的 12 个省或领域，而且也包含了作为总的加拿大自身。这时，我们可能需要将总的加拿大排除出去，只对 12 个省或领域进行分析。其中一个方法就是将分析限制于人口少于 2 000 万人的地方，也就是说分析将排除作为全体的加拿大：

```
. summarize if pop < 20000
Variable |     Obs        Mean    Std. Dev.       Min        Max
---------+-----------------------------------------------------
   place |       0
     pop |      12    2467.158    3435.521       30.1    11100.3
   unemp |      10       12.26     4.44877          7       19.6
   mlife |      12      74.225    1.728965       70.2       75.8
   flife |      12    80.68333      1.0116         78       81.8
```

比较这里与前面的 **summarize** 结果就可以看出有多大的变化了。比如，前面取得的人口平均数存在着极大的误导性，因为它将所有的人都计数 2 次。

"<"（表示小于）是一种关系运算符，一共有 6 种关系运算符（relation operators），说明如下：

==	等于
!=	不等于　（也可以用 ~=）
>	大于
<	小于
>=	大于等于
<=	小于等于

两个连续等号"=="标志一种逻辑检验，表示"是否左侧的值与右侧的值相等"。对于 Stata 而言，一个等号则代表了另外的意思，它表示"让左侧的值与右侧的值相同"。

单一等号不是关系运算符,也根本不能用于 **if** 选择条件内。单一等号有另外的意义。它们用于根据代数表达式来创建一个新变量,或者将计算值替代原来的变量值。单一等号也用于某些特殊应用场合,比如,加权和假设检验。

所有这些关系运算符都能用于按照数值变量的观测值来选择案例。只有两种关系运算符"=="和"!="只对字符串变量有意义。在对字符串变量应用 **if** 选择条件时,要将目标值括在英文双引号中。比如,我们可以用下面的命令取得除整体加拿大以外(只留其12个省和领域)的概要统计:

```
. summarize if place != "Canada"
```

通过加入逻辑运算符便可以在一个 **if** 选择中包含两个或更多的关系运算符。Stata的逻辑运算符有以下几种:

& 和

| 或 (注意,这是个符号,而不是数字1或字母l)

! 否 (也可以用~表示)

加拿大领域中 Yukon 和 Northwest 的人口都少于100 000人。要想排除这两个最小的地方和最大的地方(加拿大整体),只输出另外10个省的失业率和预期寿命的平均数,我们可以应用以下命令:

```
. summarize unemp mlife flife if pop > 100 & pop < 20000
Variable |       Obs        Mean    Std. Dev.       Min        Max
---------+--------------------------------------------------------
   unemp |        10       12.26     4.44877          7       19.6
   mlife |        10       74.92    .6051633       73.9       75.8
   flife |        10       80.98     .586515       79.8       81.8
```

采用括号可以定义多重运算符的优先顺序。比如,我们可以列出所有失业率低于9或男性预期寿命高于75.4和女性预期寿命高于81.4的地方:

```
. list if unemp < 9 | (mlife >= 75.4 & flife >= 81.4)
     +------------------------------------------------------+
     |            place      pop   unemp   mlife   flife |
     |------------------------------------------------------|
  8. |         Manitoba   1137.5     8.5      75    80.8 |
  9. |     Saskatchewan   1015.6       7    75.2    81.8 |
 10. |          Alberta     2747     8.4    75.5    81.4 |
 11. | British Columbia     3766     9.8    75.8    81.4 |
     +------------------------------------------------------+
```

关于缺失值的注意事项:Stata 一般将缺失值显示为一个点,但在某些运算中(特别是在 **sort** 和 **if** 运算中,尽管在统计计算如平均数或相关不是这样),这些相同的缺失值被当作非常大的正值。如果我们按地区的失业率由小到大排序,然后要求查看哪些地方的失业率高于15%,看看会发生什么事情:

```
. sort unemp

. list if unemp > 15
     +-----------------------------------------------------------+
     |                 place     pop   unemp   mlife   flife |
     |-----------------------------------------------------------|
 10. |  Prince Edward Island   136.1    19.1    74.8    81.3 |
 11. |          Newfoundland   575.4    19.6    73.9    79.8 |
 12. |                 Yukon    30.1       .    71.3    80.4 |
 13. | Northwest Territories    65.8       .    70.2      78 |
     +-----------------------------------------------------------+
```

注意两个失业率缺失的地方也被列在其中,成了"大于15"的地方。在这种情况下,问题是明显的,但是要是对于很大的数据而言,我们就有可能发现不了。假设我们分析一个

政治选举的民意测验结果。下列命令就会将变量 *vote* 列表,不仅是按照预想包括了 65 以上的人,而且还会包括所有年龄值缺失者:

```
. tabulate vote if age > 65
```

由于存在数据缺失者,我们就不得不明确地在 **if** 表达式中对此加以处理。

```
. tabulate vote if age > 65 & age < .
```

像"**age < .**"的小于不等式是一种选择非缺失值案例的通用方法。尽管这里我们只使用了默认的"."缺失值码,其实 Stata 还允许设立 27 种不同的缺失值码。其他 26 种码在系统内代表着比"."更大的数值,所以 **<.** 便会排除所有这些缺失情况。键入 **help missing** 可以取得更多的相应细节。

选择条件 **in** 和 **if** 只是将观测案例暂时地置于操作之外,以便某一个命令并不应用于它们。这些选择条件并不影响内存中的数据,下一条命令就会应用于所有的观测案例,除非其再次用了 **in** 和 **if** 选择。如果想要从内存中清除数据中的变量,需要采用 **drop** 命令。比如,要从内存中清除 *mlife* 和 *flife*,可以键入:

```
. drop mlife flife
```

我们还能通过 **in** 选择或 **if** 选择从内存中清除观测案例。由于我们在前面曾经按 *unemp* 排过序,上述两个领域居于数据的第 12 项和第 13 项。加拿大本身处于第 6 项。可以应用 **in** 选择来清除这三个不是省的地域。命令 **drop in 12 /13** 意味着"清除第 12 至 13 项观测案例"。

```
. list

     +----------------------------------------+
     |                place       pop   unemp |
     |----------------------------------------|
  1. |         Saskatchewan    1015.6       7 |
  2. |              Alberta      2747     8.4 |
  3. |             Manitoba    1137.5     8.5 |
  4. |              Ontario   11100.3     9.3 |
  5. |     British Columbia      3766     9.8 |
     |----------------------------------------|
  6. |               Canada   29606.1    10.6 |
  7. |               Quebec    7334.2    13.2 |
  8. |        New Brunswick     760.1    13.8 |
  9. |          Nova Scotia     937.8    13.9 |
 10. | Prince Edward Island     136.1    19.1 |
     |----------------------------------------|
 11. |         Newfoundland     575.4    19.6 |
 12. |                Yukon      30.1       . |
 13. | Northwest Territories     65.8       . |
     +----------------------------------------+

. drop in 12/13
(2 observations deleted)

. drop in 6
(1 observation deleted)
```

同样的改变也可以通过执行 **if** 选择来完成,这一命令表示"如果 *place* 为加拿大或人口少于 100 便加以清除"。

```
. drop if place == "Canada" | pop < 100
(3 observations deleted)
```

清除了加拿大和其他领域并清除了变量 *mlife* 和 *flife* 以后,我们便有了以下简化的数据:

```
. list
```

```
     +-------------------------------------+
     |                 place     pop  unemp |
     |-------------------------------------|
  1. |          Saskatchewan  1015.6      7 |
  2. |               Alberta    2747    8.4 |
  3. |              Manitoba  1137.5    8.5 |
  4. |               Ontario 11100.3    9.3 |
  5. |      British Columbia    3766    9.8 |
     |-------------------------------------|
  6. |                Quebec  7334.2   13.2 |
  7. |         New Brunswick   760.1   13.8 |
  8. |           Nova Scotia   937.8   13.9 |
  9. |  Prince Edward Island   136.1   19.1 |
 10. |          Newfoundland   575.4   19.6 |
     +-------------------------------------+
```

我们也可以用数据编辑器的 **Delete** 按钮来清除所选的变量或观测案例。

除了告诉 Stata 哪些变量或观测案例需要清除外，有时我们指定哪些变量或观测案例需要保留则更为简单。上述同样的数据简化也可以通过以下命令来取得：

```
. keep place pop unemp

. keep if place != "Canada" & pop >= 100
(3 observations deleted)
```

如同其他对内存数据的改变一样，这些简化都不会影响到磁盘上的文件，除非我们将这些数据进行存盘。这时，我们有一个选择是覆盖旧的数据文件（**save, replace**），于是就摧毁了旧文件；另一个选择是将新修改的数据存为一个新命名的文件（选择 **File-Save As** 或者键入形式为 **save *newname*** 的命令），这样一来两个版本的数据都会存在于磁盘上。

创建和替代变量

命令 **generate** 和 **replace** 使我们可以创建新的变量或者改变现有变量的值。比如，如同大多数工业国家一样，加拿大的女性比男性的寿命要长。为了分析这一性别差异上的地区差别，我们可以调用数据 *canada1.dta*，并创建一个新变量为女性预期寿命（*flife*）与男性预期寿命（*mlife*）之差。在命令 **generate** 或 **replace** 的主要部分，我们使用单一等号符（这与 **if** 选择不同）。

```
. use canada1, clear
(Canadian dataset 1)

. generate gap = flife - mlife

. label variable gap "Female-male gap life expectancy"

. describe
Contains data from C:\data\canada1.dta
  obs:            13                          Canadian dataset 1
 vars:             6                          3 Jul 2005 10:48
 size:           585 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
place           str21  %21s                   Place name
pop             float  %9.0g                  Population in 1000s, 1995
unemp           float  %9.0g                  % 15+ population unemployed,
                                                1995
mlife           float  %9.0g                  Male life expectancy years
flife           float  %9.0g                  Female life expectancy years
gap             float  %9.0g                  Female-male gap life expectancy
-------------------------------------------------------------------------------
Sorted by:
```

```
. list place flife mlife gap

     +---------------------------------------------------+
     |                 place   flife   mlife         gap |
     |---------------------------------------------------|
  1. |                Canada    81.1    75.1           6 |
  2. |          Newfoundland    79.8    73.9    5.900002 |
  3. |  Prince Edward Island    81.3    74.8         6.5 |
  4. |           Nova Scotia    80.4    74.2    6.200005 |
  5. |         New Brunswick    80.6    74.8    5.799995 |
     |---------------------------------------------------|
  6. |                Quebec    81.2    74.5    6.699997 |
  7. |               Ontario    81.1    75.5    5.599998 |
  8. |              Manitoba    80.8      75    5.800003 |
  9. |          Saskatchewan    81.8    75.2    6.600006 |
 10. |               Alberta    81.4    75.5    5.900002 |
     |---------------------------------------------------|
 11. |      British Columbia    81.4    75.8    5.599998 |
 12. |                 Yukon    80.4    71.3    9.099998 |
 13. | Northwest Territories      78    70.2    7.800003 |
     +---------------------------------------------------+
```

对于省份纽芬兰(Newfoundland),新变量 *gap* 的真实数值应该为 79.8-73.9 = 5.9岁,但是输出中却显示这个值为 5.900 002。与其他计算机程序一样,Stata 以二进制形式存放数据,而 5.9 并没确切的二进制代表。这一小小的不准确起源于二进制中对小数部分的近似,不会对统计计算有什么影响,因为计算是以双精度(每个数占 8 字节)进行的。然而,它们显示在数据表中令人不安。我们可以改变显示格式,让 Stata 按四舍五入方式显示。以下命令指定了固定的显示格式,总列宽为 4 位,显示 1 位小数:

```
. format gap %4.1f
```

然而,就是这个数值显示为 5.9 了,以下命令却输出不了相应的观测案例:

```
. list if gap == 5.9
```

这是因为 Stata 相信这个数值并不正好等于 5.9。(从技术角度讲,Stata 是以单精度存放 *gap* 数值然而却以双精度进行所有计算,并且 5.9 的单精度近似值与双精度近似值并不相同)

显示格式,以及变量名和标签,也可以在数据编辑器里双击相应列来完成。固定数字格式如 **% 4.1f** 只是三种最常用的数字显示格式类型之一。这些显示格式类型是:

- % *w.d***g** 一般(general)数字格式,其中 *w* 定义了数字显示宽度或占几列,而 *d* 定义了小数部分至少要显示的位数。为了以最佳(但是可变)方式来显示,指数记数法(比如,1.00e+7,表示 1.00×10^7 或 1 000 万)和小数点位置移动都会按需要自动完成。
- % *w.d***f** 固定(fixed)数字格式,其中 *w* 定义了数字显示的总宽度,而 *d* 定义了小数部分的固定显示位数。
- % *w.d***e** 指数(exponential)数字格式,其中 *w* 定义了数字显示的总宽度,而 *d* 定义了小数部分的固定显示位数。

比如,在表 2.1 中我们看到,加拿大 1995 年的人口近似为 29 606 100 人,而 Yukon领域人口为 30 100 人。让我们看看这两个数据在几种不同显示格式(format)下是怎样的:

format	Canada	Yukon
%9.0g	2.96e+07	30100
%9.1f	29606100.0	30100.0
%12.5e	2.96061e+07	3.01000e+04

尽管所显示的数值看起来很不同,其实它们的内部数值是相同的。统计计算并不受到显示格式的影响。其他数字显示格式选项还包括用逗号分隔、左对齐和右对齐、左侧空位补0。此外还有日期、时间序列变量和字符串变量的特殊格式。请参阅 **help format** 提供的更多信息。

命令 **replace** 可以完成 **generate** 一样的各类计算,但是它不是创建一个新的变量,而是替代一个现有变量的数值。比如,在我们的数据中变量 *pop* 是以千为单位提供人口数值的。如果要将其转变为简单的人口数,我们只需要将所有的值乘以1 000即可(命令中的"*"即表示乘以)。

```
. replace pop = pop * 1000
```

命令 **replace** 可以做这种大规模改变,也可以与 **in** 或 **if** 条件一起使用来选择性地编辑数据。为了示范,假设我们有包括年龄 *age* 和出生年份 *born* 这两个变量的问卷数据。以下命令可以改正那些年龄为29却被输入为229的错误:

```
. replace age = 29 if age == 229
```

此外,下面的命令可以对第1 453条观测案例的年龄值的错误进行修改:

```
. replace age = 29 in 1453
```

再举一个更复杂的例子,

```
. replace age = 2005-born if age >= .  |  age < 2005-born
```

如果在 *age* 缺失或报告的年龄小于2005减去出生年的差值时,这条命令将用2005减去出生年得到的差值来取代原来的age变量值。

命令 **generate** 和 **replace** 还提供了创建分类变量的工具。前面我们注意到加拿大数据中包含着几种不同类型的观测案例:有2个领土地域,10个省,还有1个整个国家的观测案例。尽管用 **in** 和 **if** 条件可以用来选择,并且用 **drop** 也能清除数据,最方便的作法可能还是设置一个分类变量来表示观测案例的"类型"。以下我们示范一种方法来建立这样一个变量 *type*。先创建一个常数值的 *type*,各条观测案例都赋值为1。然后,我们将 Yukon 和 Northwest Territories 的 *type* 值替换为2,将Canada的 *type* 值替换为3。最后的一步工作是为这个新变量制作标签,并且定义变量值1、2、3的标签。

```
. use canada1, clear
(Canadian dataset 1)
. generate type = 1
. replace type = 2 if place == "Yukon" | place == "Northwest
      Territories"
(2 real changes made)
. replace type = 3 if place == "Canada"
(1 real change made)
. label variable type "Province, territory or nation"
. label values type typelbl
. label define typelbl 1 "Province" 2 "Territory" 3 "Nation"
. list place flife mlife gap type
```

```
     +--------------------------------------------------------------------+
     |                    place   flife   mlife        gap         type |
     |--------------------------------------------------------------------|
  1. |                   Canada    81.1    75.1          6       Nation |
  2. |             Newfoundland    79.8    73.9   5.900002     Province |
  3. |     Prince Edward Island    81.3    74.8        6.5     Province |
  4. |              Nova Scotia    80.4    74.2   6.200005     Province |
  5. |            New Brunswick    80.6    74.8   5.799995     Province |
     |--------------------------------------------------------------------|
  6. |                   Quebec    81.2    74.5   6.699997     Province |
  7. |                  Ontario    81.1    75.5   5.599998     Province |
  8. |                 Manitoba    80.8      75   5.800003     Province |
  9. |             Saskatchewan    81.8    75.2   6.600006     Province |
 10. |                  Alberta    81.4    75.5   5.900002     Province |
     |--------------------------------------------------------------------|
 11. |         British Columbia    81.4    75.8   5.599998     Province |
 12. |                    Yukon    80.4    71.3   9.099998    Territory |
 13. |    Northwest Territories      78    70.2   7.800003    Territory |
     +--------------------------------------------------------------------+
```

正如所示，为分类变量做标签需要两条命令。命令 **label define** 指定哪个标签与哪些数值相联系。而命令 **label values** 指定的是这些标签与哪个变量相联系。一套标签(用命令 **label define** 建立)可以应用于任何数量的变量(即在 **label values** 命令中可以指定许多变量来参照)。变量值标签可以包括32 000 个字符，但是在它们不太长时各种任务便能工作得最好。

命令 **generate** 能够使用任何老变量、常数、随机值和表达式的任意组合来创建新的变量，**replace** 则能够为老变量产生新的值。对于数值变量而言，可以应用以下代数运算符：

+	加
-	减
*	乘
/	除
^	乘方

用括号来控制计算的顺序。当没有括号时，计算将采用通常的优先顺序。对于代数运算符，只有加法，即“ + ”，可以用于字符串变量，功能是将两个字符串连接成为一个。

尽管目的不同，**generate** 和 **replace** 有类似的命令语法。它们都能采用 Stata 运算符和 **in** 与 **if** 选择上的任何数学或逻辑的可能组合。这些命令还可以运用 Stata 大批的特殊函数，我们将在下一节对此加以介绍。

使用函数

这一节介绍许多与 **generate** 或 **replace** 一起使用的函数。比如，我们要创建一个名为 *loginc* 的新变量，等于收入变量 *income* 的自然对数，那么我们就要在 **generate** 命令中使用自然对数函数 **ln**。

```
. generate loginc = ln(income)
```

自然对数 **ln** 只是 Stata 数学函数之一。下面列出这些函数：

abs(x)	x 的绝对值
acos(x)	反余弦函数。因为 360 度 =2 π 弧度，**acos**(x) ***180 / _pi** 得到反余弦的反算度数(其中 **_pi** 表示数学上的常数 π)
asin(x)	反正弦函数
atan(x)	反正切函数

atan2(*y*,*x*)　　y/x 的反正切函数

atanh(*x*)　　双曲反正切函数

ceil(*x*)　　大于等于 *x* 的最小整数,比如,在 $n-1<x\leqslant n$ 时得整数 *n*

cloglog(*x*)　　*x* 的互补双对数:即 ln(-ln(1-x))

comb(*n*,*k*)　　组合函数(*n* 个中一次取 *k* 个时所有可能组合数)

cos(*x*)　　余弦函数。要知道 *y* 度的余弦,键入

```
generate y=cos(y *_pi /180)
```

digamma(*x*)　　$d\ln\Gamma(x)/dx$

exp(*x*)　　指数函数

floor(*x*)　　小于等于 *x* 的最大整数,比如,在 $n\leqslant x<n+1$ 时得整数 *n*

trunc(*x*)　　截取 *x* 的整数部分

invcloglog(*x*)　　*x* 的互补双对数的逆:即 1-exp(-exp(x))

invlogit(*x*)　　*x* 的 logit 转换的逆:即 exp(x)/(1+exp(x))

ln(*x*)　　*x* 的自然对数(以 *e* 为底)。计算任何其他以数字 B 为底的 x 的对数,可键入命令

```
generate y= ln(x)/ln(B)
```

lnfactorial(*x*)　　*x* 的阶乘的自然对数。求 x 的阶乘,可键入命令

```
generate y= round(exp(lnfact(x),1))
```

lngamma(*x*)　　Γ(x)的自然对数。求 Γ(x),可键入命令

```
generate y= exp(lngamma(x))
```

log(*x*)　　类似 ln(*x*)的自然对数

$\mathbf{log_{10}}$(*x*)　　*x* 的对数(以 10 为底)

logit(*x*)　　*x* 的对数发生比:即 ln(*x*/(1-*x*))

max(*x1*,*x2*,...,*xn*)　　*x1*,*x2*,...,*xn* 中的最大值

min(*x1*,*x2*,...,*xn*)　　*x1*,*x2*,...,*xn* 中的最小值

mod(*x*,*y*)　　与 *x*/*y* 的余数。

reldif(*x*,*y*)　　相对差异:即 $|x-y|/(|y|+1)$

round(*x*)　　*x* 的四舍五入整数

round(*x*,*y*)　　按 *y* 为单位 *x* 四舍五入

sign(*x*)　　符号函数:当 $x<0$ 时为 -1,当 $x=0$ 时为 0,当 $x>0$ 时为 +1

sin(*x*)　　正弦函数

sqrt(*x*)　　平方根函数

total(*x*)　　*x* 的移动合计(参见 **help egen**)

tan(*x*)　　正切函数

tanh(*x*)　　双曲正切函数

trigamma(*x*)　　即 $d^2\ln\Gamma(x)/dx^2$

还有许多概率函数可用,下面也予以列出。有关重要细节,包括定义、参数的限制条件和缺失值的处理,请参见 **help probfun** 和文献手册。

betaden(*a*,*b*,*x*)　　贝塔分布的概率密度函数

Binomial(*n*,*k*,*p*)　　二项分布:在一次试验的成功概率为 *p* 的条件下,在 *n* 次试验中有 *k* 次及更多次成功的概率

binormal(*h*,*k*,*r*)　　双正态分布:两个相关系数为 *r* 的正态分布的联合累计分布

函数	说明
chi2(*n*,*x*)	自由度为 *n* 的累计卡方分布
chi2tail(*n*,*x*)	自由度为 *n* 的反向累计(右侧,存活)卡方分布: chi2tail(*n*,*x*) =1-chi2(*n*,*x*)
dgammapda(*a*,*x*)	累计伽玛分布的 gammap(*a*,*x*)对 *a* 的偏导数
dgammapdx(*a*,*x*)	累计伽玛分布的 gammap(*a*,*x*)对 *x* 的偏导数
dgammapdada(*a*,*x*)	累计伽玛分布的 gammap(*a*,*x*)对 *a* 的二阶偏导数
dgammapdadx(*a*,*x*)	累计伽玛分布的 gammap(*a*,*x*)对 *a* 和 *x* 的二阶偏导数
dgammapdxdx(*a*,*x*)	累计伽玛分布的 gammap(*a*,*x*)对 *x* 的二阶偏导数
F(*n1*,*n2*,*f*)	分子、分母自由度分别为 *n1* 和 *n2* 的累计 *F* 分布
Fden(*n1*,*n2*,*f*)	分子、分母自由度分别为 *n1* 和 *n2* 的 *F* 分布的概率密度
Ftail(*n1*,*n2*,*f*)	分子、分母自由度分别为 *n1* 和 *n2* 的反向累计(上端,存活)*F* 分布 Ftail(*n1*,*n2*,*f*) = 1 -F(*n1*,*n2*,*f*)
gammaden(*a*,*b*,*g*,*x*)	伽玛族分布的概率密度函数,其中 gammaden(*a*,1,0,*x*) = 累计伽玛分布 gammap(*a*,*x*)的概率密度
gammap(*a*,*x*)	对于 *a* 的累计伽玛分布;又称为不完整伽玛分布
ibeta(*a*,*b*,*x*)	对于 *a* 和 *b* 的累计伽玛分布;又称为不完整伽玛分布
invbinomial(*n*,*k*,*P*)	二项分布逆运算。当 $P \le 0.5$ 时,本函数求一次试验的成功概率 *p*,使 *n* 次试验中有 *k* 次及以上成功的概率为 *P*;当 $P > 0.5$ 时,概率 *p* 使 *n* 次试验中有 *k* 次及以下成功的概率为 $1 - P$
invchi2(*n*,*p*)	卡方分布逆运算。如果 **chi2**(*n*,*x*) =*p*,有 invchi2(*n*,*p*) =*x*
invchi2tail(*n*,*p*)	**chi2tail**( )的逆运算。如果 chi2tail(*n*,*x*) =*p*,有 invchi2tail(*n*,*p*) =*x*
invF(*n1*,*n2*,*p*)	累计 *F* 分布的逆运算。如果 F(*n1*,*n2*,*f*) =*p*,有 invF(*n1*,*n2*,*p*) =*f*
invFtail(*n1*,*n2*,*p*)	反向累计 *F* 分布的逆运算。如果 Ftail(*n1*,*n2*,*f*) = *p*,有 invFtail(*n1*,*n2*,*p*) =*f*
invgammap(*a*,*p*)	累计伽玛分布的逆运算。如果 gammap(*a*,*x*) =*p*,有 invgammap(*a*,*p*) =*x*
invibeta(*a*,*b*,*p*)	累计贝塔分布的逆运算。如果 ibeta(*a*,*b*,*x*) =*p*,有 invibeta(*a*,*b*,*p*) =*x*
invnchi2(*n*,*L*,*p*)	累计非中心卡方分布的逆运算。如果 nchi2(*n*,*L*,*x*) = *p*,有 invnchi2(*n*,*L*,*p*) =*x*
invnFtail(*n1*,*n2*,*L*,*p*)	反向累计非中心 *F* 分布的逆运算。如果 nFtail(*n1*,*n2*,*L*,*f*) =*p*,有 invnFtail(*n1*,*n2*,*L*,*p*) =*f*
invnibeta(*a*,*b*,*L*,*p*)	累计非中心贝塔分布的逆运算。如果 nibeta(*a*,*b*,*L*,*x*) =*p*,有 invnibeta(*a*,*b*,*L*,*p*) =*x*
invnormal(*p*)	累计标准正态分布的逆运算。如果 normal(*z*) =*p*,

	有 invnormal(p) = Z
invttail(n,p)	反向累计 t 分布的逆运算。如果 ttail(n,t) = p,有 invttail(n,p) = t
nbetaden(a,b,L,x)	非中心伽玛分布的概率密度函数,有形状参数 a 和 b 以及非中心参数 L
nchi2(n,L,x)	累计非中心卡方分布,其自由度为 n 以及非中心参数 L
nFden($n1,n2,L,x$)	非中心 F 分布密度,其自由度分别为 $n1$ 和 $n2$,非中心参数为 L
nFtail($n1,n2,L,x$)	反向累计(上端,存活)非中心 F 分布,自由度分别为 $n1$ 和 $n2$,非中心参数为 L
nibeta(a,b,L,x)	累计非中心伽玛分布,形状参数为 a 和 b,非中心参数为 L
normal(z)	累计标准正态分布
normalden(z)	标准正态分布密度,平均数为 0,标准差为 1
normalden(z,s)	正态分布密度,平均数为 0,标准差为 s
normalden(x,m,s)	正态分布密度,平均数为 m,标准差为 s
npnchi2(n,x,p)	求累计非中心卡方分布的非中心参数值。如果 nchi2(n,L,x) = p,有 npnchi2(n,x,p) = L
tden(n,t)	自由度为 n 的 t 分布密度
ttail(n,t)	自由度为 n 的反向累计(上端)t 分布。这一函数反求 $T>t$ 的概率
uniform( )	伪随机数据发生器,获得区间[0,1)内理论均匀分布的返回值

在 **uniform**( )的括号内无参数。根据需要,我们可以控制这一伪随机数发生器的初始值、乃至整个系列的"随机"数。相应命令为 **set seed #**,其中#可以是从 0 到 $2^{31}-1$之间的任意整数。省略 **set seed** 命令就对应着要求 **set seed 123456789**,于是总是产生同样的系列数。

Stata 还提供 40 多种日期函数以及与日期相关的时间序列函数。在其《用户指南》的第 27 章提供了有关清单,或者通过键入 **help datefun** 来查询。下面提供了一些日期函数的例子。这些函数中所谓的"消逝天数"指自从 1960 年 1 月 1 日起已经过了多少天。

date(s_1, s_2[,**y**])	就 s_1 的消逝天数。s_1 实际上是个任意格式的表示日期的字符串变量。可以应用月份,缩写为三个字母,或者用数字表示。也可以用年份表示,可以包含世纪、也可以不包含世纪;包括空格和标点也都允许。s_2 是关于月(m)、日(d)、年([##]y)是如何在 s_1 中排列顺序的定义。##为 s_1 中两位数年份所属的世纪,其默认格式为 19y。
d(*l*)	这个函数是为了使日期表达更方便。比如,键入**d(2jan 1960**)与直接键入 1 结果一样。
mdy(***m***,***d***,***y***)	对应日期 m,d,y 的消逝天数。
day(***e***)	对应消逝日期 e 为相应月份的第几天。

month(*e*)	对应消逝日期 *e* 的月份。
year(*e*)	对应消逝日期 *e* 的年份。
dow(*e*)	对应消逝日期 *e* 为相应周中的第几天。
doy(*e*)	对应消逝日期 *e* 为相应年份的第几天。
week(*e*)	对应消逝日期 *e* 为相应年份的第几周。
quarter(*e*)	对应消逝日期 *e* 为相应年份的第几季度。
halfyear(*e*)	对应消逝日期 *e* 为相应年份的哪个半年。

另外,还有一些特殊函数也很有用,将其列在下面:

autocode(*x*,*n*,*xmin*,*xmax*)	根据 *x* 值形成分类变量:将 *x* 的值域(即最小值 *xmin* 至最大值 *xmax*)分成等距的 *n* 份,并求出各 *x* 值所在区间的上限。
cond(*x*,*a*,*b*)	当评价 *x* 值时为"肯定"时返回 *a* 值,评价 *x* 值时为"否定"时返回 *b* 值。比如: `. generate y = cond(inc1 > inc2, inc1, inc2)` 形成变量 *y*,其值取 *inc1* 与 *inc2* 中的最大值(假定无缺失值)。
group(*x*)	建立一个分类变量,将按排序后的数据分为尽量等规模的 *x* 个子样本。
trunc(*x*)	求 *x* 平截(truncate,即删除其小数部分)后的整数。
max(x_1,x_2,...,x_n)	求 x_1,x_2,...,x_n中的最大值。忽略其中缺失值。比如,**max(3 +2,1)**将求得 5。
min(x_1,x_2,...,x_n)	求 x_1,x_2,...,x_n中的最小值。
recode(x,x_1,x_2,...,x_n)	当 *x* 缺失时求得缺失值,当 $x < x_1$ 时求得 x_1,当 $x < x_2$ 时求得 x_2。
round(*x*,*y*)	以 *y* 为单位对 *x* 做四舍五入。
sign(*x*)	符号函数:当 $x < 0$ 时得-1,当 $x = 0$ 时得 0,当 $x > 0$时得 +1。(当 *x* 缺失时返回缺失值)
total(*x*)	*x* 的移动合计,将缺失值作为 0 对待。

字符串函数(这里不再描述)用于处理和评价字符串变量。请键入 **help strfun** 查看字符串变量的全部清单。参考手册和《用户指南》中给出了示例与这些以及其他函数的详细说明。

如果需要,多重的函数、运算符和选择条件可以在一个命令中组合起来。以上描述的函数和代数运算也可以用于其他不创建或修改数据变量的工作。命令 **display** 执行单一计算并且将结果显示在屏幕上。比如:

```
. display 2+3
5

. display log10(10^83)
83

. display invttail(120,.025) * 34.1/sqrt(975)
2.1622305
```

于是,**display** 可以作为屏幕显示的统计计算器来用。

与计算器不同,**display**、**generate** 和 **replace** 可以直接得到 Stata 的统计结果。比如,假设我们要依据数据集 *canada1.dta* 汇总失业率统计:

```
. summarize unemp

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       unemp |        11    12.10909    4.250048          7       19.6
```

在 **summarize** 之后,Stata 将平均数作为一个名为 r(mean) 的宏临时保存下来。

```
. display r(mean)
12.109091
```

我们可以使用这一结果来创建一个变量 *unempDEV*,表示对平均数的离差(deviation)。

```
. gen unempDEV = unemp - r(mean)
(2 missing values generated)

. summ unemp unempDEV

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       unemp |        11    12.10909    4.250048          7       19.6
    unempDEV |        11    4.33e-08    4.250048  -5.109091    7.49091
```

Stata 还提供了另一个创建变量的命令,**egen**(表示是对 **generate** 命令的扩展,即"extensions to generate"),它有自己的一系列用于完成 **generate** 命令无法轻易完成的函数。这些函数包括计算现有变量或变量表达式的总和、最大值、最小值、中位数、四分位距、标准化分值或移动平均数等,依据这些计算来创建新变量。比如,下述命令创建了一个名为 *zscore* 的新变量,它等于 *x* 的标准化(平均数为 0,方差为 1)分值:

```
. egen zscore = std(x)
```

再比如,下述命令创建了一个名为 *avg* 的新变量,它等于每一观测案例在 *x*, *y*, *z* 和 *w* 四个变量上的忽略了任何缺失值的行平均数。

```
. egen avg = rowmean(x,y,z,w)
```

为了创建一个名为 *sum* 的新变量,它等于每一观测案例在 *x*,*y*,*z* 和 *w* 四个变量上的合计,并将四个变量中的缺失值视为零,键入:

```
. egen sum = rowsum(x,y,z,w)
```

下述命令创建一个名为 *xrank* 的新变量,它保持着与 *x* 取值相一致的排序:对于 *x* 取大值的观测案例,*xrank* =1;对于取第二大值的观测案例,*xrank* =2,如此等等。

```
. egen xrank = rank(x)
```

通过键入 **help egen** 可以得到 **egen** 函数的完整清单,或者查阅有更多举例的参考手册。

数值和字符串之间的格式转换

数据集 *canada2.dta* 包含一个字符串变量 *place*。它还包含一个已经添加了取

值标签的分类变量,*type*。它们似乎都具有非数值的取值。

```
. use canada2, clear
(Canadian dataset 2)

. list place type

     +-----------------------------------+
     |                 place        type |
     |-----------------------------------|
  1. |                Canada      Nation |
  2. |          Newfoundland    Province |
  3. |  Prince Edward Island    Province |
  4. |           Nova Scotia    Province |
  5. |         New Brunswick    Province |
     |-----------------------------------|
  6. |                Quebec    Province |
  7. |               Ontario    Province |
  8. |              Manitoba    Province |
  9. |          Saskatchewan    Province |
 10. |               Alberta    Province |
     |-----------------------------------|
 11. |      British Columbia    Province |
 12. |                 Yukon   Territory |
 13. | Northwest Territories   Territory |
     +-----------------------------------+
```

其实,在标签之下,*type* 仍然是数值型变量,下面我们一增加 **nolabel** 选项之后就能看到:

```
. list place type, nolabel

     +------------------------------+
     |                 place   type |
     |------------------------------|
  1. |                Canada      3 |
  2. |          Newfoundland      1 |
  3. |  Prince Edward Island      1 |
  4. |           Nova Scotia      1 |
  5. |         New Brunswick      1 |
     |------------------------------|
  6. |                Quebec      1 |
  7. |               Ontario      1 |
  8. |              Manitoba      1 |
  9. |          Saskatchewan      1 |
 10. |               Alberta      1 |
     |------------------------------|
 11. |      British Columbia      1 |
 12. |                 Yukon      2 |
 13. | Northwest Territories      2 |
     +------------------------------+
```

字符串变量和添加了取值标签的数值型变量看起来很像,但是在分析中它们的表现却不同。对于字符串变量而言,大多数统计运算和代数关系都不能应用,因此我们可能想要在数据中同时包括反映同一信息的字符串变量和添加了取值标签的数值型变量。**encode** 命令可以依据字符串变量创建一个添加了取值标签的数值型变量。数字 1 被赋给字符串按字母顺序排在第一位的那个,随后是 2,如此等等。下例中,我们依据字符串变量 *place* 创建了一个名为 *placenum* 的添加了取值标签的数值型变量:

```
. encode place, gen(placenum)
```

相反的转换也是可能的:**decode** 命令可以使用添加了取值标签的数值型变量的值创建字符串变量。这里,我们根据数值型变量 *type* 创建字符串变量 *typestr* :

```
. decode type, gen(typestr)
```

把数据列出来就可以看到,新的数值型变量 *placenum* 和字符串变量 *typestr* 与原先的变量在显示上一样:

```
. list place placenum type typestr

     +------------------------------------------------------------------------+
     |                  place                  placenum        type   typestr |
     |------------------------------------------------------------------------|
  1. |                 Canada                    Canada      Nation    Nation |
  2. |           Newfoundland              Newfoundland    Province  Province |
  3. |   Prince Edward Island      Prince Edward Island    Province  Province |
  4. |            Nova Scotia               Nova Scotia    Province  Province |
  5. |          New Brunswick             New Brunswick    Province  Province |
     |------------------------------------------------------------------------|
  6. |                 Quebec                    Quebec    Province  Province |
  7. |                Ontario                   Ontario    Province  Province |
  8. |               Manitoba                  Manitoba    Province  Province |
  9. |           Saskatchewan              Saskatchewan    Province  Province |
 10. |                Alberta                   Alberta    Province  Province |
     |------------------------------------------------------------------------|
 11. |       British Columbia          British Columbia    Province  Province |
 12. |                  Yukon                     Yukon   Territory Territory |
 13. | Northwest Territories     Northwest Territories   Territory Territory |
     +------------------------------------------------------------------------+
```

但是,如果加上 **nolabel** 选项的话,差异就显现出来了。Stata 是将 *placenum* 和 *type* 当作数值型变量对待的。

```
. list place placenum type typestr, nolabel

     +-------------------------------------------------------------------+
     |                  place               placenum   type    typestr |
     |-------------------------------------------------------------------|
  1. |                 Canada   3.00000000000000e+00      3     Nation |
  2. |           Newfoundland   6.00000000000000e+00      1   Province |
  3. |   Prince Edward Island   1.00000000000000e+01      1   Province |
  4. |            Nova Scotia   8.00000000000000e+00      1   Province |
  5. |          New Brunswick   5.00000000000000e+00      1   Province |
     |-------------------------------------------------------------------|
  6. |                 Quebec   1.10000000000000e+01      1   Province |
  7. |                Ontario   9.00000000000000e+00      1   Province |
  8. |               Manitoba   4.00000000000000e+00      1   Province |
  9. |           Saskatchewan   1.20000000000000e+01      1   Province |
 10. |                Alberta   1.00000000000000e+00      1   Province |
     |-------------------------------------------------------------------|
 11. |       British Columbia   2.00000000000000e+00      1   Province |
 12. |                  Yukon   1.30000000000000e+01      2  Territory |
 13. | Northwest Territories   7.00000000000000e+00      2  Territory |
     +-------------------------------------------------------------------+
```

诸如平均数和标准差等统计分析只能应用于数值型变量。就计算的目的而言,有无数值型变量的标签(对计算结果)无关紧要。

```
. summarize place placenum type typestr

Variable |      Obs        Mean    Std. Dev.        Min         Max
---------+-----------------------------------------------------------
   place |        0
placenum |       13           7     3.89444           1          13
    type |       13    1.307692    .6304252           1           3
 typestr |        0
```

有时,我们会遇到字符串变量的取值全部或绝大部分都为数字的情况。为了将这些字符取值转换成与其相对应的数字,可以使用 **real** 函数。比如,尽管下述 *siblings* 变量仍属于字符串变量,其实只有"4 或更多"(4 or more)这一种取值可能不大容易用一个数字加以表示。

```
. describe siblings

  1. siblings     str9    %9s          Number of siblings (string)

. list
```

```
     +-----------+
     |  siblings |
     |-----------|
  1. |         0 |
  2. |         1 |
  3. |         2 |
  4. |         3 |
  5. | 4 or more |
     +-----------+
```

```
. generate sibnum = real(siblings)
(1 missing value generated)
```

新变量 *sibnum* 属于数值型变量，当 *siblings* 为“4 或更多”时，其取值为缺失。

```
. list

     +--------------------+
     |  siblings   sibnum |
     |--------------------|
  1. |         0        0 |
  2. |         1        1 |
  3. |         2        2 |
  4. |         3        3 |
  5. | 4 or more        . |
     +--------------------+
```

destring 命令提供了将字符串变量转换成数值型变量的更灵活的方法。对上面的例子，我们可以通过键入下述命令做到同样的事情：

```
. destring siblings, generate(sibnum) force
```

有关该命令和选项的信息，请见 **help destring**。

创建新的分类变量和定序变量

上一节说明了如何创建一个名为 *type* 的分类变量来对加拿大数据中的领域、省份和全国案例加以区分。用户可以采用许多其他方法来创建分类或定序变量。本节将给出一些示例。

type 变量包括三个类别：

```
. tabulate type

Province,    |
territory or |
nation       |      Freq.     Percent        Cum.
-------------+-----------------------------------
    Province |         10       76.92       76.92
   Territory |          2       15.38       92.31
      Nation |          1        7.69      100.00
-------------+-----------------------------------
       Total |         13      100.00
```

考虑到某些需要，我们可能想将这一多分类的变量重新表达成一组编码分为 1 或 0 的二分变量或“虚拟变量”(dummy variables)。如果加上 **generate** 选项的话，**tabulate** 可以自动创建一组虚拟变量。在下面的例子中，形成了一组名为 *type1*、*type2* 和 *type3* 的变量，每一个变量代表 *type* 变量三类中的一类：

```
. tabulate type, generate(type)

Province,    |
territory or |
nation       |      Freq.     Percent        Cum.
-------------+-----------------------------------
    Province |         10       76.92       76.92
   Territory |          2       15.38       92.31
      Nation |          1        7.69      100.00
-------------+-----------------------------------
       Total |         13      100.00
```

```
. describe

Contains data from C:\data\canada2.dta
  obs:            13                          Canadian dataset 2
 vars:            10                          3 Jul 2005 10:48
 size:           637 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
place           str21  %21s                   Place name
pop             float  %9.0g                  Population in 1000s, 1995
unemp           float  %9.0g                  % 15+ population unemployed,
                                                1995
mlife           float  %9.0g                  Male life expectancy years
flife           float  %9.0g                  Female life expectancy years
gap             float  %9.0g                  Female-male gap life expectancy
type            byte   %9.0g       typelbl    Province, territory or nation
type1           byte   %8.0g                  type==Province
type2           byte   %8.0g                  type==Territory
type3           byte   %8.0g                  type==Nation
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

. list place type type1-type3

     +-------------------------------------------------------------+
     |                 place        type   type1   type2   type3 |
     |-------------------------------------------------------------|
  1. |                Canada      Nation       0       0       1 |
  2. |          Newfoundland    Province       1       0       0 |
  3. |  Prince Edward Island    Province       1       0       0 |
  4. |           Nova Scotia    Province       1       0       0 |
  5. |         New Brunswick    Province       1       0       0 |
     |-------------------------------------------------------------|
  6. |                Quebec    Province       1       0       0 |
  7. |               Ontario    Province       1       0       0 |
  8. |              Manitoba    Province       1       0       0 |
  9. |          Saskatchewan    Province       1       0       0 |
 10. |               Alberta    Province       1       0       0 |
     |-------------------------------------------------------------|
 11. |      British Columbia    Province       1       0       0 |
 12. |                 Yukon   Territory       0       1       0 |
 13. | Northwest Territories   Territory       0       1       0 |
     +-------------------------------------------------------------+
```

将分类变量的信息重新表达成一组虚拟变量并不会出现信息损失；本例中，由 *type1* 到 *type3* 共同准确地提供了与 *type* 本身同样多的信息。有时候，尽管会造成信息的大量损失，分析人员还是选择将测量型变量转换为分类或定序的格式。比如，*canada2 .dta* 数据中的 *unemp* 变量提供了失业率测量。排除数据中的加拿大自身之后，我们看到 *unemp* 的分布在7% 到19.6%，平均数为12.26。

```
. summarize unemp if type != 3

Variable |       Obs        Mean    Std. Dev.       Min        Max
---------+--------------------------------------------------------
   unemp |        10       12.26     4.44877          7       19.6
```

在这一意义上，数据中的加拿大案例成为一种干扰，因此我们将它清除：

```
. drop if type == 3
(1 observation deleted)
```

用两条命令来创建一个名为 *unemp2* 的虚拟变量：当 *unemp* 低于平均水平(12.26)时，令 *unemp2* 等于0；当 *unemp* 等于或高于平均水平时，令 *unemp2* 等于1；而当 *unemp* 为缺失值时，令 *unemp2* 也为缺失值。在读到第二条命令时，请记住 Stata 的排序和关系运算符将缺失值作为极大的数字对待。

```
. generate unemp2 = 0 if unemp < 12.26
(7 missing values generated)
```

```
. replace unemp2 = 1 if unemp >= 12.26 & unemp < .
(5 real changes made)
```

我们可能想就某个测量变量的取值进行分组，从而创建一个有序的分类变量，即定序变量。**autocode** 函数（请参见本章前面的“使用函数”一节）提供了变量的自动分组功能。为了创建一个新的定序变量 *unemp3*，使它将 *unemp* 从 5 到 20 的取值区间分成等宽的三组，请键入：

```
. generate unemp3 = autocode(unemp,3,5,20)
(2 missing values generated)
```

列出该数据，可以看到新的虚拟变量（*unemp2*）和定序变量（*unemp2*）是如何与原始测量变量 *unemp* 的取值相对应的。

```
. list place unemp unemp2 unemp3

     +-----------------------------------------------+
     |                place   unemp   unemp2   unemp3 |
     |-----------------------------------------------|
  1. |         Newfoundland    19.6        1       20 |
  2. |  Prince Edward Island   19.1        1       20 |
  3. |          Nova Scotia    13.9        1       15 |
  4. |        New Brunswick    13.8        1       15 |
  5. |               Quebec    13.2        1       15 |
     |-----------------------------------------------|
  6. |              Ontario     9.3        0       10 |
  7. |             Manitoba     8.5        0       10 |
  8. |         Saskatchewan       7        0       10 |
  9. |              Alberta     8.4        0       10 |
 10. |     British Columbia     9.8        0       10 |
     |-----------------------------------------------|
 11. |                Yukon       .        .        . |
 12. | Northwest Territories      .        .        . |
     +-----------------------------------------------+
```

刚才提到的两种策略都恰当地处理了缺失值，因此在 *unemp* 上为缺失值的那些案例在由 *unemp* 转换的变量也为缺失值。如果数据中没有缺失值，另外一种可能的转换方法更好。为了说明这点，我们先要删除 Yukon 和 Northwest 这两个地区：

```
. drop if unemp >= .
(2 observations deleted)
```

诸如 ***unemp* > = .**等表示大于等于的不等式除了选择默认缺失码“.”之外，还会选择所有由用户定义的缺失码。请键入 **help missing** 查看细节。

删除完具有缺失值的观测案例，我们现在可以使用 **group** 函数来创建一个定序变量，该变量不是像 **autocode** 那样为近似等宽的分组，而是为近似等规模的分组。我们分两步来实现这点。第一，基于关注的变量对数据进行排序（假定没有缺失值）。第二，使用 **group**(#)函数创建一个新变量，其中的#表示期望的分组数目。下述例子将加拿大的 10 个省份分为 5 个组。

```
. sort unemp
. generate unemp5 = group(5)
. list place unemp unemp2 unemp3 unemp5
```

```
     +------------------------------------------------------------+
     |                place   unemp   unemp2   unemp3   unemp5 |
     |------------------------------------------------------------|
  1. |         Saskatchewan       7        0       10        1 |
  2. |              Alberta     8.4        0       10        1 |
  3. |             Manitoba     8.5        0       10        2 |
  4. |              Ontario     9.3        0       10        2 |
  5. |     British Columbia     9.8        0       10        3 |
     |------------------------------------------------------------|
  6. |               Quebec    13.2        1       15        3 |
  7. |        New Brunswick    13.8        1       15        4 |
  8. |          Nova Scotia    13.9        1       15        4 |
  9. | Prince Edward Island    19.1        1       20        5 |
 10. |         Newfoundland    19.6        1       20        5 |
     +------------------------------------------------------------+
```

另一差别在于,**autocode** 分配的数值等于每一区间的上界,而 **group** 只是简单地将第一组赋值为 1,第二组为 2,如此等等。

标注变量下标

当 Stata 有数据在内存中时,它也定义了描述这些数据的系统变量。比如,*_N* 表示观测案例总数。*_n* 表示观测案例号:*_n* = 1 表示第一条观测案例,*_n* = 2 表示第二条观测案例,如此等等,直到最后一条观测案例(*_n* = *_N*)。如果我们键入如下命令,就会创建一个新变量 *caseID*,其值等于前面已经排序过的每一条观测案例的序号。

```
. generate caseID = _n
```

如果按其他方式对数据排序将会改变每一观测案例的 *_n* 值,但是其 *caseID* 的取值将保持不变。因此,如果我们以不同方式对数据排序,以后再键入下述命令就能恢复原来的顺序:

```
. sort caseID
```

创建并保存数据集形成初期观测案例的唯一性顺序识别码能够大大便利以后的数据管理。

我们能够对变量名添加下标来指定独特的观测案例的号码。比如,数据集 *canada1.dta* 中的第 6 条观测案例(如果我们没有删除任何记录或者没有进行重新排序)是 Quebec。因此,*pop*[6]指的是 Quebec 的人口数,7 334 千人。

```
. display pop[6]
7334.2002
```

类似地,*pop*[12]便是 Yukon 的人口数。

```
. display pop[12]
30.1
```

当我们的数据构成一个序列时,加注下标和_n 系统变量具有另外的好处。比如,如果我们以某支股票每天的股市价格作为名为 *price* 的变量,那么 *price* 或者等价的 *price*[_n]表示第 n 次观测或第 n 天的价格,*price*[_n-1]表示前一天的价格,而 *price*[_n+1]则表示后一天的价格。因此,我们可以定义一个新变量 *difprice*,它等于自前一天来的价格变化:

```
. generate difprice = price - price[_n-1]
```

有关时间序列分析的第 13 章会讨论这一主题。

导入其他程序的数据

前面几节介绍了如何直接在数据编辑器中用键盘录入和编辑数据。如果我们的数据保存在恰当编排格式的电子表格中,有一些捷径可以加快这一工作过程:我们可以直接拷贝电子表格中的多列数据块(不包含列标签),然后粘贴到 Stata 的数据编辑器中。这需要很小心,有时甚至需要先做试验,因为 Stata 会将任何包含了非数字取值的列作为字符串变量处理。文本或文字处理文档中单独的一列数据(变量)也可以粘贴到 Stata 中。一旦数据被成功地粘贴到编辑器的各列中,就可以采用常规方式来指定变量名、标签等。

这些数据编辑器方法都快捷而简单,但是对于大型数据而言,就得有专门工具来直接处理由其他程序创建的数据文件。这些文件大体上可分成两类:一种是原始数据的 ASCII(文本)文件,这类数据可以采用恰当的 Stata 命令读入到 Stata 中;另一种是系统文件,这类数据必须通过特定的第三方程序转换成 Stata 格式后 Stata 才能读入。

为了示范读入 ASCII 文件的方法,我们回到表 2.1 的加拿大数据。假如不是将这些数据直接键入 Stata 数据编辑器,而是先将它们键入到其他文字处理器中,并且每个值之间至少空一格。如果字符串内包含空格,就必须加上双引号,比如,"Prince Edward Island"。对于其他的字符型取值,引号可有可无。文字处理器可以将文档存为 ASCII(文本)格式文件,这种格式比一般文字处理软件的存取格式更简单、而且更为通用。因此,我们可以创建一个如下形式的名为 *canada.raw* 的 ASCII 文件:

```
"Canada"  29606.1  10.6  75.1  81.1
"Newfoundland"  575.4  19.6  73.9  79.8
"Prince Edward Island"  136.1  19.1  74.8  81.3
"Nova Scotia"  937.8  13.9  74.2  80.4
"New Brunswick"  760.1  13.8  74.8  80.6
"Quebec"  7334.2  13.2  74.5  81.2
"Ontario"  11100.3  9.3  75.5  81.1
"Manitoba"  1137.5  8.5  75  80.8
"Saskatchewan"  1015.6  7  75.2  81.8
"Alberta"  2747  8.4  75.5  81.4
"British Columbia"  3766  9.8  75.8  81.4
"Yukon"  30.1  .  71.3  80.4
"Northwest Territories"  65.8  .  70.2  78
```

特别需要注意,在最后两行中,要采用英文句点,而不是空格来表示 Yukon 和 Northwest 地区的缺失值。如果数据集原本应当有五个变量,那么每一观测案例则必须正好有五个值(包括表示缺失值的句点)。

命令 **infile** 能够把诸如 *canada.raw* 这样的 ASCII 数据读入到内存中,这些数据值是由一个或多个分隔符分隔开的,分隔符可以采用空格、制表符、换行符(包括回车、换行或同时回车换行)以及英文逗号。这个命令的基本格式为(其中 ***variable-list*** 代表变量清单):

```
. infile variable-list using filename.raw
```

当全部为数值型变量时,变量清单可以省略,此时 Stata 会依次将变量命名为 *var1*、*var2*、*var3* 等。但是,我们通常可能想给每个变量取一个与众不同的名称。有时,我们还需要区别那些字符串变量。对于 *canada.raw* 的 **infile** 命令可以是:

```
. infile str30 place pop unemp mlife flife using canada.raw, clear
(13 observations read)
```

infile 的变量清单指定了变量在数据文件中出现的次序。**clear** 选项指定在读入新文件之前将内存中的所有当前数据删除。

如果数据中包含字符串变量，那么每一字符串变量名称前面都要加上 **str#** 进行说明。比如，上述命令中的 **str30** 就是告诉 Stata 下一个命名变量(*place*)是一个长度为 30 个字符的字符串变量。实际上，没有任何一个加拿大的地区名称长度超过了 21 个字符，但是我们不需要事先知道这点。通常为了方便，总是高估字符串变量的长度。因此，一旦数据已读入到内存中，使用 **compress** 来确保没有变量多占用了所需的空间。**compress** 命令会自动改变所有变量以达到最有效的内存占用存储类型。

```
. compress
place was str30 now str21

. describe
Contains data
  obs:            13
 vars:             5
 size:           533 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
place           str21  %21s
pop             float  %9.0g
unemp           float  %9.0g

mlife           float  %9.0g
flife           float  %9.0g
-------------------------------------------------------------------------------
Sorted by:
```

现在，我们可以按如前所述的方法进一步给变量和数据加上标签。在任何情况下，**save** ***canada0***（或 **save** ***canada0*** **, replace**)命令都将把 Stata 格式的新数据存成名为 *canada0.dta* 文件。最初的原始数据文件 *canada.raw* 仍然原封不动地保存在磁盘上。

如果我们的数据中包含了一些非数量值(比如，"男"和"女")，我们又想将其转换为带标签的数值型变量加以保存，那么增加 **automatic** 选项可以实现这一点。比如，我们可以用以下 **infile** 命令读入原始调查数据：

```
. infile gender age income vote using survey.raw, automatic
```

电子表格和数据库程序一般都写出每一行只有一条记录且采用制表符或英文逗号分隔的 ASCII 文件。为了将这种数据读入 Stata，需要使用 **insheet** 命令。其一般的语法和 **infile** 类似，同时带有告诉 Stata 这一数据的分隔符是制表符、逗号还是其他字符的选项。比如，假设数据是以制表符分隔，命令为：

```
. insheet variable-list using filename.raw, tab
```

或者，假设数据以逗号分隔，并且文件的第一行为变量名称(也以逗号分隔)，就键入：

```
. insheet variable-list using filename.raw, comma names
```

使用 **insheet** 命令时，我们不需要专门识别字符串变量。如果我们没有纳入变量清单并且数据文件的第一行也没有包含变量名称，Stata 会自动指定变量名称为 *var1*、*var2*、*var3* 等。如果 ASCII 文件中的一些取值并不是由 **insheet** 命令中指定的分隔符分隔的，数据读入就会出错。

其他统计软件创建的粗数据(raw data)文件可以是“固定列”格式的,其中各值之间根本不需要进行分隔,但是必须占事先确定的列位。**infile** 命令和更为专门的 **infix** 命令都允许 Stata 读取此类数据文件。要么在命令语法本身中,要么在一个以独立文件存在或者作为数据文件第一部分的“数据字典”中,我们必须准确地指定应当如何逐列读取这些数据。

这里有一个简单的例子。数据保存在一个名为 *nfresour.raw* 的 ASCII 文件中:

```
198624087641691000
198725247430001044
198825138637481086
198925358964371140
1990    8615731195
1991    7930001262
```

这些数据是关于加拿大纽芬兰省(Newfoundland)的自然资源产量的信息。四个变量占用了固定的列位置:1 ~4 列是年份(1986…1991);5 ~8 列为以千立方米度量的森林资源量(2408…缺失);9 ~14 列为以千美元度量的矿山资源量(764169…793000);15 ~18 列为相对于1986 年的消费者价格指数(1000…1262)。请注意,不同于采用空格或制表符作为分隔符的文件,在固定列位格式的数据中,空白表示缺失值,并且这一原始数据不含小数。为了把 *nfresour.raw* 读入到 Stata 中,我们指定每个变量所占的列位:

```
. infix year 1-4 wood 5-8 mines 9-14 CPI 15-18
      using nfresour.raw, clear
(6 observations read)

. list

     +------------------------------+
     | year   wood    mines    CPI  |
     |------------------------------|
  1. | 1986   2408   764169   1000  |
  2. | 1987   2524   743000   1044  |
  3. | 1988   2513   863748   1086  |
  4. | 1989   2535   896437   1140  |
  5. | 1990      .   861573   1195  |
     |------------------------------|
  6. | 1991      .   793000   1262  |
     +------------------------------+
```

更为复杂的固定列位格式数据可能需要一个数据“字典”。数据字典可以简单明了,但是它们提供了许多可能的选择。键入 **help infix** 或者 **help infile2** 获取这些命令的简要描述。更多的示例和解释,请咨询《用户指南》和参考手册。Stata 也可以加载、输出或者查看来自 ODBC(Open Database Connectivity)资源的数据;请参见 **help odbc**。

如果我们需要将数据从 Stata 输出到其他的非 ODBC 程序,那该怎么办? **outfile** 命令可将 ASCII 文件写到磁盘上。下述命令将创建一个名为 *canada6 .raw* 的以空格为分隔符的 ASCII 文件,该文件包含了内存中的所有数据信息:

```
. outfile using canada6
```

上述的 **infile**、**insheet**、**infix** 和 **outfile** 命令都针对以 ASCII 文件保存的原始数据进行操作。另一个非常快捷的办法是从 Stata 数据浏览器中拷贝数据并直接将其粘贴到诸如 Excel 等电子数据表中。但是,最好的选择往往还是在不同的数据表、数据库或统计程序存储的特殊系统文件之间直接进行数据转换。有好几种第三方程序

能够做这种翻译。比如,Stat/Transfer 可以在许多不同格式数据之间进行转换,包括 dBASE、Excel、FoxPro、Gauss、JMP、Lotus、MATLAB、Minitab、OSIRIS、Paradox、S-Plus、SAS、SPSS、SYSTAT 和 Stata。该软件通过 Stata 公司(www.stata.com)或者从其生产者 Circle Systems (www.stattransfer.com)那里可以获得。对于在多程序环境中工作或需要与同事交换数据的分析人员而言,这种转换程序提供了不可或缺的工具。

合并两个或多个 Stata 文件

我们可以采用两种一般方法来合并 Stata 数据集: **append**(附加)第二个包含其他观测案例的数据集;或者和其他包含新变量或取值的数据文件进行 **merge**(合并)。为保持与本章加拿大例子相一致,我们将使用有关纽芬兰省(Newfoundland)的数据来示范这些操作程序。文件 *newf1.dta* 记录了纽芬兰省从 1985 年到 1989 年的人口数。

```
. use newf1, clear
(Newfoundland 1985-89)

. describe

Contains data from C:\data\newf1.dta
  obs:             5                          Newfoundland 1985-89
 vars:             2                          3 Jul 2005 10:49
 size:            50 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %9.0g                  Year
pop             float  %9.0g                  Population
-------------------------------------------------------------------------------
Sorted by:

. list

     +---------------+
     | year      pop |
     |---------------|
  1. | 1985   580700 |
  2. | 1986   580200 |
  3. | 1987   568200 |
  4. | 1988   568000 |
  5. | 1989   570000 |
     +---------------+
```

文件 *newf2.dta* 包含了随后若干年的人口数和失业人数信息。

```
. use newf2
(Newfoundland 1990-95)

. describe

Contains data from C:\data\newf2.dta
  obs:             6                          Newfoundland 1990-95
 vars:             3                          3 Jul 2005 10:49
 size:            84 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %9.0g                  Year
pop             float  %9.0g                  Population
jobless         float  %9.0g                  Number of people unemployed
-------------------------------------------------------------------------------
Sorted by:
```

```
. list

     +-------------------------+
     | year      pop   jobless |
     |-------------------------|
  1. | 1990   573400     42000 |
  2. | 1991   573500     45000 |
  3. | 1992   575600     49000 |
  4. | 1993   584400     49000 |
  5. | 1994   582400     50000 |
     |-------------------------|
  6. | 1995   575449         . |
     +-------------------------+
```

为了合并这两个数据,考虑到 *newf2.dta* 已经读入内存中,我们使用 **append** 命令:

```
. append using newf1
. list

     +-------------------------+
     | year      pop   jobless |
     |-------------------------|
  1. | 1990   573400     42000 |
  2. | 1991   573500     45000 |
  3. | 1992   575600     49000 |
  4. | 1993   584400     49000 |
  5. | 1994   582400     50000 |
     |-------------------------|
  6. | 1995   575449         . |
  7. | 1985   580700         . |
  8. | 1986   580200         . |
  9. | 1987   568200         . |
 10. | 1988   568000         . |
     |-------------------------|
 11. | 1989   570000         . |
     +-------------------------+
```

因为变量 *jobless* 出现在 *newf2* (1990—1995 年)但没有出现在 *newf1* 中,因此合并后的数据中该变量从 1985 年到 1989 年各年份为缺失值。我们现在可以将这些观测案例按照时间先后排序并把该合并数据另存成名为 *newf3.dta* 的新文件:

```
. sort year
. list

     +-------------------------+
     | year      pop   jobless |
     |-------------------------|
  1. | 1985   580700         . |
  2. | 1986   580200         . |
  3. | 1987   568200         . |
  4. | 1988   568000         . |
  5. | 1989   570000         . |
     |-------------------------|
  6. | 1990   573400     42000 |
  7. | 1991   573500     45000 |
  8. | 1992   575600     49000 |
  9. | 1993   584400     49000 |
 10. | 1994   582400     50000 |
     |-------------------------|
 11. | 1995   575449         . |
     +-------------------------+

. save newf3
```

通过将另一个包含新观测案例(行)的文件添加到一个文件的底部,**append** 可被看作是将该文件(也就是,内存中的数据集)加长。从最简单的形式上看,通过将另一个文件添加到一个文件的右边从而增加新变量(列),**merge** 相当于将该文件"加宽"。比如,数据集 *newf4.dta* 进一步包含纽芬兰省的时间序列信息:1980—1994 年期间的出生数和离婚数。因此其中有一些观测案例以及一个变量(*year*)和我们前面的数据集

*newf3 .dta*是共同的，但它还包含两个在 *newf3 .dta* 中没有出现的新变量。

```
. use newf4
(Newfoundland 1980-94)

. describe

Contains data from C:\data\newf4.dta
  obs:            15                          Newfoundland 1980-94
 vars:             3                          3 Jul 2005 10:49
 size:           150 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %9.0g                  Year
births          int    %9.0g                  Number of births
divorces        int    %9.0g                  Number of divorces
-------------------------------------------------------------------------------
Sorted by:

. list

     +--------------------------+
     | year   births   divorces |
     |--------------------------|
  1. | 1980    10332        555 |
  2. | 1981    11310        569 |
  3. | 1982     9173        625 |
  4. | 1983     9630        711 |
  5. | 1984     8560        590 |
     |--------------------------|
  6. | 1985     8080        561 |
  7. | 1986     8320        610 |
  8. | 1987     7656       1002 |
  9. | 1988     7396        884 |
 10. | 1989     7996        981 |
     |--------------------------|
 11. | 1990     7354        973 |
 12. | 1991     6929        912 |
 13. | 1992     6689        867 |
 14. | 1993     6360        930 |
 15. | 1994     6295        933 |
     +--------------------------+
```

我们想将 *newf3* 和 *newf4* 进行合并，并根据变量 *year* 对同一年份的观测案例进行匹配。为了做到这点，两个数据集都必须根据索引变量(index variable)(本例中为 *year*)进行排序。我们已经在保存 *newf3 .dta* 之前执行过命令 **sort year** 命令，因此现在只需对*newf4 .dta*做同样的事情。然后，指定 *year* 作为匹配时的索引变量，我们就可以将两个数据加以合并。

```
. sort year

. merge year using newf3

. describe

Contains data from newf4.dta
  obs:            16                          Newfoundland 1980-94
 vars:             6                          3 Jul 2005 10:49
 size:           304 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %9.0g                  Year
births          int    %9.0g                  Number of births
divorces        int    %9.0g                  Number of divorces
pop             float  %9.0g                  Population
jobless         float  %9.0g                  Number of people unemployed
_merge          byte   %8.0g
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

```
. list

     +------------------------------------------------------+
     | year   births   divorces      pop   jobless   _merge |
     |------------------------------------------------------|
  1. | 1980    10332        555        .         .        1 |
  2. | 1981    11310        569        .         .        1 |
  3. | 1982     9173        625        .         .        1 |
  4. | 1983     9630        711        .         .        1 |
  5. | 1984     8560        590        .         .        1 |
     |------------------------------------------------------|
  6. | 1985     8080        561   580700         .        3 |
  7. | 1986     8320        610   580200         .        3 |
  8. | 1987     7656       1002   568200         .        3 |
  9. | 1988     7396        884   568000         .        3 |
 10. | 1989     7996        981   570000         .        3 |
     |------------------------------------------------------|
 11. | 1990     7354        973   573400     42000        3 |
 12. | 1991     6929        912   573500     45000        3 |
 13. | 1992     6689        867   575600     49000        3 |
 14. | 1993     6360        930   584400     49000        3 |
 15. | 1994     6295        933   582400     50000        3 |
     |------------------------------------------------------|
 16. | 1995        .          .   575449         .        2 |
     +------------------------------------------------------+
```

在本例中,我们只是用 **merge** 按观测案例的匹配将新变量添加到我们的数据中。在默认状态下,当两个数据集存在相同的变量时,"主"(master)数据(即内存中的文件)中的那些被保留下来,"调用"(using)数据中的那些则被忽略。但是,**merge** 命令有几个选项可以更改这一默认状态。以下命令将允许主数据中出现的缺失值由调用数据(即 *newf5 .dta*)中相应的非缺失值进行替换。

```
. merge year using newf5, update
```

或者,用以下命令可使主数据中的任何取值如与调用数据存在不同时将由后者的非缺失值进行替换:

```
. merge year using newf5, update replace
```

假如索引变量的某种取值在主数据中出现多次,比如,1990 年出现两次,那么调用数据中 *year* 取值为 1990 年的观测案例将会与主干数据中每一条 *year* 为 1990 年的观测案例进行匹配。用户可以利用这种能力来实现许多意图,比如,将每个病人的背景信息与其每次不同的医生就诊的信息加以合并。尽管 **merge** 使得此类和许多其他的数据管理任务变得很简单,但是分析人员应当认真查看所得结果以确认该命令所得结果正是所想要的。

作为一种诊断辅助,**merge** 会自动创建一个名为 *_merge* 的新变量。除非设定了 **update**,否则 *_merge* 编码的含义如下:

1.观测案例只来自于主数据。

2.观测案例只来自于调用数据。

3.观测案例同时来自于主数据和调用数据(如果出现不同,忽略调用数据值)。如果设定了 **update** 选项, *_merge* 编码会指示发生了什么:

1.观测案例只来自于主数据。

2.观测案例只来自于调用数据。

3.观测案例同时来自于主数据和调用数据,且主数据和调用数据一致。

4.观测案例同时来自于主数据和调用数据,如果主数据为缺失值,则被更新。

5.观测案例同时来自于主数据和调用数据,如果出现不同,主数据将被替换。

在执行另一 **merge** 操作之前,必须删除该变量或改变该变量的名称。比如:

```
. drop _merge
```

或者,

```
. rename _merge _merge1
```

我们可以用一个 **merge** 命令来合并多个数据。比如,如果从 *newf5.dta* 到*newf8.dta*为四个数据集,每一个都根据变量 *year* 进行了排序,那么将这四个数据合并到主数据中的命令如下:

```
. merge year using newf5 newf6 newf7 newf8, update replace
```

其他的 **merge** 选项还包括核对合并变量的取值是否唯一和指定哪些变量保留在最终数据中。具体细节请键入 **help merge** 进行查询。

数据的转置、变换或分析

数据集创建起来之后,我们可能发现该数据的结构对于某些分析目的而言是错误的。很幸运,有几条命令方便了数据结构的改变。我们将使用加拿大五个东部省份近年来的人口增长数据(*growth1.dta*)来对此做示范。和前面的例子不同,这些数据中的省份名称由数值型变量表示,并对变量编制了最多 8 个字符组成的取值标签。

```
. use growth1, clear
(Eastern Canada growth)

. describe

Contains data from C:\data\growth1.dta
  obs:             5                          Eastern Canada growth
 vars:             5                          3 Jul 2005 10:48
 size:           105 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
provinc2        byte   %8.0g       provinc2   Eastern Canadian province
grow92          float  %9.0g                  Pop. gain in 1000s, 1991-92
grow93          float  %9.0g                  Pop. gain in 1000s, 1992-93
grow94          float  %9.0g                  Pop. gain in 1000s, 1993-94
grow95          float  %9.0g                  Pop. gain in 1000s, 1994-95
-------------------------------------------------------------------------------
Sorted by:

. list

     +----------------------------------------------------+
     | provinc2   grow92   grow93   grow94   grow95 |
     |----------------------------------------------------|
  1. | New Brun       10      2.5      2.2      2.4 |
  2. | Newfound      4.5       .8       -3     -5.8 |
  3. | Nova Sco     12.1      5.8      3.5      3.9 |
  4. |  Ontario    174.9    169.1    120.9    163.9 |
  5. |   Quebec     80.6     77.4     48.5     47.1 |
     +----------------------------------------------------+
```

在这一数据中,每年的人口增长被分别作为变量加以存储。我们可以分析各年人口增长平均数或方差的变化。但是这一给定的数据结构却使 Stata 不能轻易地画出人口增长对年份的简单时间标绘图,也不能计算两个省(如 New Brunswick 和 Newfoundland)人口增长之间的相关关系。尽管这个数据已经包含了所有必要的信息,但是上述分析却要求不同的数据结构。

简单的数据结构重组涉及变量和观测案例的转置。实际上,就是使数据中的行变成

列,反之亦然。这可以通过 **xpose** 命令来实现。这一命令要求必须加上 **clear** 选项,因为它总是会从内存中清除当前的数据。增加 **varname** 选项可在转置后的数据中创建一个附加的变量(被命名为*_varname*),用以包含作为字符串的原始变量名。

```
. xpose, clear varname

. describe

Contains data
  obs:            5
 vars:            6
 size:          160 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
v1              float  %9.0g
v2              float  %9.0g
v3              float  %9.0g
v4              float  %9.0g
v5              float  %9.0g
_varname        str8   %9s
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

. list

     +-----------------------------------------------+
     |  v1     v2     v3      v4     v5   _varname |
     |-----------------------------------------------|
  1. |   1      2      3       4      5   provinc2 |
  2. |  10    4.5   12.1   174.9   80.6     grow92 |
  3. | 2.5     .8    5.8   169.1   77.4     grow93 |
  4. | 2.2     -3    3.5   120.9   48.5     grow94 |
  5. | 2.4   -5.8    3.9   163.9   47.1     grow95 |
     +-----------------------------------------------+
```

在转置中变量取值标签会丢失,因此转置后数据中的省份只是由相应的数字来指示(1 = New Brunswick,2 = Newfoundland,等等)。每一列中第二到最后一个数值为该省的各年人口增量,以千人为单位。因此,变量 *v1* 第一行中的数值为省份识别码(1 就代表新不伦瑞克省,New Brunswick),这个省从 1992 年到 1995 年的人口增长数分别在该变量的第二行到第五行。比如,通过键入 **correlate** 命令和 **in 2 /5**(第二到第五条观测案例)这一选择条件,我们现在可以计算不同省份人口增长之间的相关了:

```
. correlate v1-v5 in 2/5
(obs=4)

             |       v1       v2       v3       v4       v5
-------------+---------------------------------------------
          v1 |   1.0000
          v2 |   0.8058   1.0000
          v3 |   0.9742   0.8978   1.0000
          v4 |   0.5070   0.4803   0.6204   1.0000
          v5 |   0.6526   0.9362   0.8049   0.6765   1.0000
```

最强的相关在临海的新不伦瑞克省(New Brunswick,即 *v1*)和新斯科舍省(Nova Scotia,即 *v3*)的增长之间:$r = 0.974\ 2$。纽芬兰省(Newfoundland,即 *v2*)和安大略省(Ontario,即 *v4*)的人口增长之间的相关要更弱得多:$r = 0.480\ 3$。

更为复杂的数据结构转换可能需要通过 **reshape**(改变形状)命令。该命令可以在被称作"宽"(wide)和"长"(long)的两种基本格式之间进行数据转换。数据集 *growth1.dta* 最初为宽格式。

```
. use growth1, clear
(Eastern Canada growth)

. list
```

```
  +------------------------------------------------+
  | provinc2    grow92    grow93    grow94   grow95 |
  |------------------------------------------------|
1. | New Brun       10       2.5       2.2      2.4 |
2. | Newfound      4.5        .8        -3     -5.8 |
3. | Nova Sco     12.1       5.8       3.5      3.9 |
4. |  Ontario    174.9     169.1     120.9    163.9 |
5. |   Quebec     80.6      77.4      48.5     47.1 |
  +------------------------------------------------+
```

reshape 命令可将其转换成长格式。

```
. reshape long grow, i(provinc2) j(year)
(note: j = 92 93 94 95)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                        5   ->     20
Number of variables                   5   ->      3
j variable (4 values)                     ->   year
xij variables:
              grow92 grow93 ... grow95   ->   grow
-----------------------------------------------------------------------------
```

列出该数据可显示出它们是如何被变换的。**list** 命令加上 **sepby**( )选项形成了以下表格,表中水平线分隔的是省份,而不是默认情况下的每五个观测案例。

```
. list, sepby(provinc2)

     +--------------------------+
     | provinc2    year    grow |
     |--------------------------|
  1. | New Brun      92      10 |
  2. | New Brun      93     2.5 |
  3. | New Brun      94     2.2 |
  4. | New Brun      95     2.4 |
     |--------------------------|
  5. | Newfound      92     4.5 |
  6. | Newfound      93      .8 |
  7. | Newfound      94      -3 |
  8. | Newfound      95    -5.8 |
     |--------------------------|
  9. | Nova Sco      92    12.1 |
 10. | Nova Sco      93     5.8 |
 11. | Nova Sco      94     3.5 |
 12. | Nova Sco      95     3.9 |
     |--------------------------|
 13. |  Ontario      92   174.9 |
 14. |  Ontario      93   169.1 |
 15. |  Ontario      94   120.9 |
 16. |  Ontario      95   163.9 |
     |--------------------------|
 17. |   Quebec      92    80.6 |
 18. |   Quebec      93    77.4 |
 19. |   Quebec      94    48.5 |
 20. |   Quebec      95    47.1 |
     +--------------------------+

. label data "Eastern Canadian growth--long"

. label variable grow "Population growth in 1000s"

. save growth2
file C:\data\growth2.dta saved
```

上述 **reshape** 命令以表明我们要将数据转换成 **long**(长)格式开头。接着,它把即将创建的新变量命名为 *grow*。选项 **i**(***provinc2***)指定观测案例的识别码(identifier),或者是指定一个取值唯一从而能够标明逻辑观测的变量。在本例中,每个省份构成了一种逻辑观测。**j**(***year***)选项指定下属观测案例的识别码,或者是指定一个(在每一逻辑观测内)取值唯一从而能够标明下属观测案例的变量。这里,每一省份的下属观测案例是不同年份。

图 2.1 展示了长格式数据的一种可能用途。我们现在使用一个 **graph** 命令便可以

画出其中三个省份(选择了 New Brunswick、Newfoundland 和 Nova Scotia,即 *provinc2* <4 的那部分观测案例)进行比较的时间标绘图。以下的 **graph** 命令要求画出 *provinc2* <4 那部分观测案例的 *grow*(作为 y 轴变量)对 *year*(x 轴)的连线图,同时设置水平线于 y = 0 处(人口零增长),并且对 *provic2* 的每个取值分别画图。

```
. graph twoway connected grow year if provinc2 < 4, yline(0)
      by(provinc2)
```

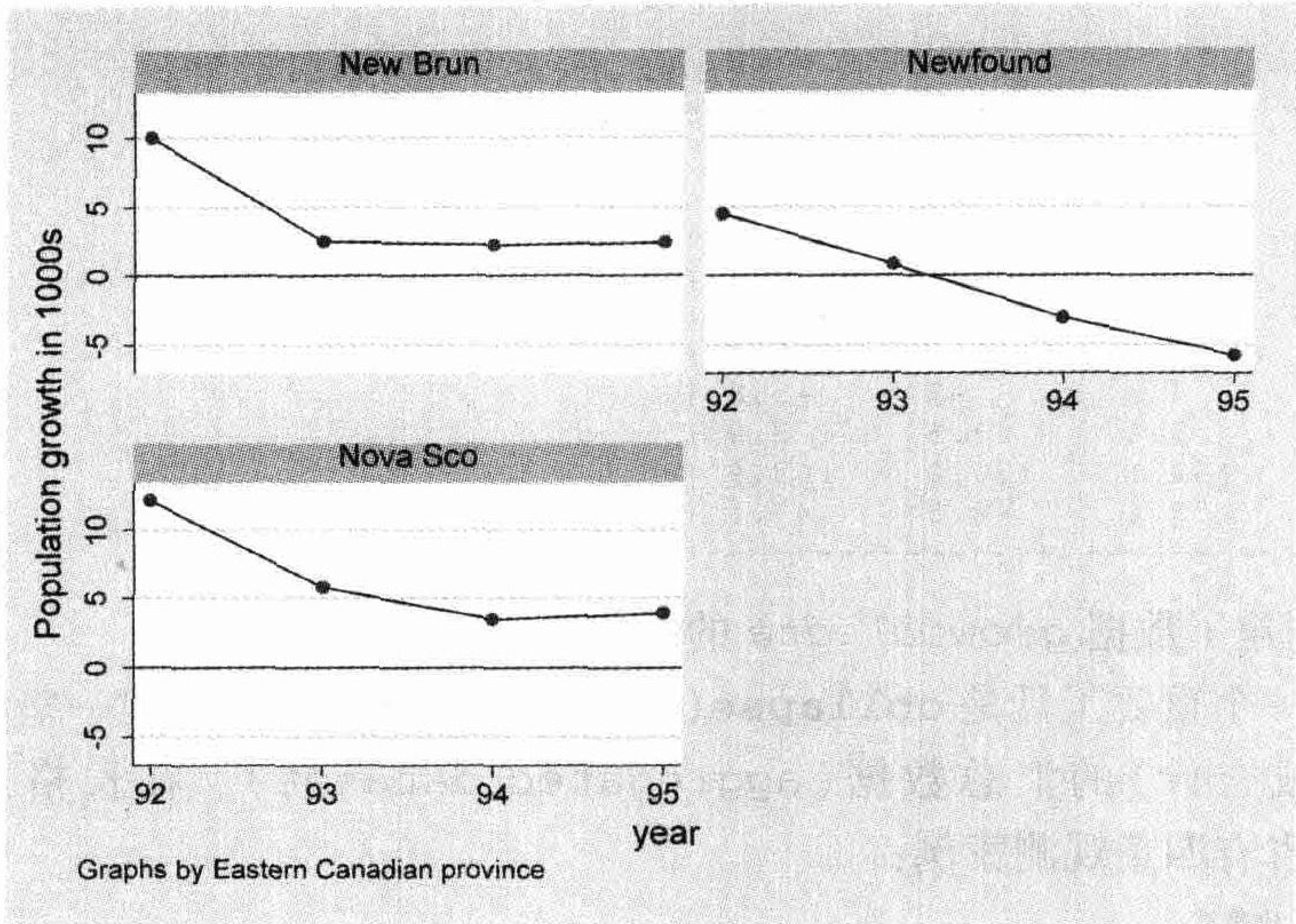


图 2.1

这三个省在 1990 年代早期渔业的衰落造成了经济困难。新不伦瑞克省(New Brun)和新斯科舍省(Nova Sco)的人口增长显著减慢,而纽芬兰省(Newfound,一个主要依赖渔业的省份)实际上出现了人口缩减。

reshape 同样也能很好地反过来用于将数据从"长"格式转换成"宽"格式。数据集 *growth3.dta* 作为长格式的一个示例。

```
. use growth3, clear
(Eastern Canadian growth--long)

. list, sepby(provinc2)
     +---------------------------+
     | provinc2    grow    year  |
     |---------------------------|
  1. | New Brun      10      92  |
  2. | New Brun     2.5      93  |
  3. | New Brun     2.2      94  |
  4. | New Brun     2.4      95  |
     |---------------------------|
  5. | Newfound     4.5      92  |
  6. | Newfound      .8      93  |
  7. | Newfound      -3      94  |
  8. | Newfound    -5.8      95  |
     |---------------------------|
  9. | Nova Sco    12.1      92  |
 10. | Nova Sco     5.8      93  |
 11. | Nova Sco     3.5      94  |
 12. | Nova Sco     3.9      95  |
     |---------------------------|
 13. |  Ontario   174.9      92  |
 14. |  Ontario   169.1      93  |
 15. |  Ontario   120.9      94  |
 16. |  Ontario   163.9      95  |
     |---------------------------|
 17. |   Quebec    80.6      92  |
 18. |   Quebec    77.4      93  |
 19. |   Quebec    48.5      94  |
 20. |   Quebec    47.1      95  |
     +---------------------------+
```

为了将该数据转换成宽格式,我们使用命令 **reshape wide**:

```
. reshape wide grow, i(provinc2) j(year)

(note: j = 92 93 94 95)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                       20   ->      5
Number of variables                   3   ->      5
j variable (4 values)              year   ->   (dropped)
xij variables:
                                   grow   ->   grow92 grow93 ... grow95
-----------------------------------------------------------------------------

. list

     +-------------------------------------------------+
     | provinc2    grow92    grow93    grow94    grow95 |
     |-------------------------------------------------|
  1. | New Brun        10       2.5       2.2       2.4 |
  2. | Newfound       4.5        .8        -3      -5.8 |
  3. | Nova Sco      12.1       5.8       3.5       3.9 |
  4. |  Ontario     174.9     169.1     120.9     163.9 |
  5. |   Quebec      80.6      77.4      48.5      47.1 |
     +-------------------------------------------------+
```

请注意,我们已经重新创建了数据 *growth1.dta* 的结构。

改变数据结构的另一个重要工具是 **collapse**(分拆)命令,它可用于创建一些统计量(如平均数、中位数或合计)的汇总数据(aggregated dataset)。在长格式的 *growth3* 数据中,每个省有四条观测案例:

```
. use growth3, clear
(Eastern Canadian growth--long)

. list, sepby(provinc2)

     +---------------------------+
     | provinc2    grow    year |
     |---------------------------|
  1. | New Brun      10      92 |
  2. | New Brun     2.5      93 |
  3. | New Brun     2.2      94 |
  4. | New Brun     2.4      95 |
     |---------------------------|
  5. | Newfound     4.5      92 |
  6. | Newfound      .8      93 |
  7. | Newfound      -3      94 |
  8. | Newfound    -5.8      95 |
     |---------------------------|
  9. | Nova Sco    12.1      92 |
 10. | Nova Sco     5.8      93 |
 11. | Nova Sco     3.5      94 |
 12. | Nova Sco     3.9      95 |
     |---------------------------|
 13. |  Ontario   174.9      92 |
 14. |  Ontario   169.1      93 |
 15. |  Ontario   120.9      94 |
 16. |  Ontario   163.9      95 |
     |---------------------------|
 17. |   Quebec    80.6      92 |
 18. |   Quebec    77.4      93 |
 19. |   Quebec    48.5      94 |
 20. |   Quebec    47.1      95 |
     +---------------------------+
```

我们可能想为每一省份汇总出不同年份的平均增长率。在分拆的数据中,每条观测案例将对应着 **by( )** 变量的一个取值,也就是某个省。

```
. collapse (mean) grow, by(provinc2)

. list
```

```
   +----------------------+
   | provinc2        grow |
   |----------------------|
1. | New Brun       4.275 |
2. | Newfound  -.8750001 |
3. | Nova Sco       6.325 |
4. |  Ontario       157.2 |
5. |   Quebec        63.4 |
   +----------------------+
```

举一个稍复杂的例子,假设我们有一个与 *growth3 .dta* 类似但还包含了变量出生数(*births*)、死亡数(*deaths*)和收入(*income*)的数据。我们想要按各省汇总出这些年的总出生数和总死亡数,以及平均收入(名为 *meaninc*)和中位收入(名为 *medinc*)的汇总数据集。如果我们不指定一个新变量的名称,比如,像上例中的 *grow* 那样,或者像这里的 *births* 和 *deaths*,那么分拆后的变量名称将与原来完全一样。

```
. collapse (sum) births deaths (mean) meaninc = income
      (median) medinc = income, by(provinc2)
```

collapse 能够根据以下概要统计量来创建变量:

mean	平均数(默认值;在未指定统计量类型的情况下使用)
sd	标准差
sum	合计
rawsum	忽略任意指定权数的合计
count	非缺失值的观测案例数
max	最大值
min	最小值
median	中位数
p1	第一百分位数
p2	第二百分位数(如此等等,直到 **p99**)
iqr	四分位距

观测案例的加权

Stata 接受四种加权(weighting)类型:

aweight	分析权数,用在加权最小二乘(WLS)回归以及类似的估计程序中
fweight	频数权数,用以对重复观测案例计数。频数权数必须是整数
iweight	重要性权数,但是"重要性"由用户自己定义
pweight	概率或抽样权数,等于观测案例根据抽样策略被选中的概率的倒数

研究者有时会提到"加权数据"(weighted data)。这可能意味着原有的抽样方案采用特意设定的非等比例方式选取观测案例,就像权数等于"1/选中概率"所反映的那样。在某些分析中,恰当使用 **pweight** 可以对非等比例抽样进行补偿。另一方面,"加权数据"可能意味着不同的东西,比如,汇总的数据集,它可能是根据一个或多个变量的频数表或交互表建构而成的,并且其中有变量表明某个特定数值或数值组合出现频数。在此种情形下,我们需要使用 **fweight**。

对于各种分析类型而言,并不是所有的加权类型都适用。比如,我们不能对 **tabulate** 命令使用 **pweight**。在任何分析中使用权数都要求我们清楚地知道在该分析中进行加权的目的。权数本身可以是数据中的任何变量。

下面用1 381名纽芬兰乡村高中学生的调查数据(*nfschool.dta*)来示范频数加权的简单应用。

```
. describe

Contains data from C:\data\nfschool.dta
  obs:             6                          Newf.school/univer.(Seyfrit 93)
 vars:             3                          3 Jul 2005 10:50
 size:            48 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
univers         byte   %8.0g       yes        Expect to attend university?
year            byte   %8.0g                  What year of school now?
count           int    %8.0g                  observed frequency
-------------------------------------------------------------------------------
Sorted by:
```

```
. list, sep(3)

     +-------------------------+
     | univers   year   count  |
     |-------------------------|
  1. |      no     10     210  |
  2. |      no     11     260  |
  3. |      no     12     274  |
     |-------------------------|
  4. |     yes     10     224  |
  5. |     yes     11     235  |
  6. |     yes     12     178  |
     +-------------------------+
```

乍一看,该数据集好像只有6个观测案例,并且当我们针对学生是否期望上大学(*univers*)和他们目前在高中就读的年级(*year*)建立交互表时,我们就得到了每一格只有一个案例的交互表。

```
. tabulate univers year

 Expect to |
    attend |
university |     What year of school now?
         ? |        10         11         12 |     Total
-----------+---------------------------------+----------
        no |         1          1          1 |         3
       yes |         1          1          1 |         3
-----------+---------------------------------+----------
     Total |         2          2          2 |         6
```

为了理解这些数据,我们需要应用频数权数。变量 *count* 给出了频数:有210名十年级学生表示不想读大学,有260名十一年级学生表示不想读大学,等等。设定[**fweight = count**]可以取得一张显示了全部1 381名学生的应答交互表。

```
. tabulate univers year [fweight = count]

 Expect to |
    attend |
university |     What year of school now?
         ? |        10         11         12 |     Total
-----------+---------------------------------+----------
        no |       210        260        274 |       744
       yes |       224        235        178 |       637
-----------+---------------------------------+----------
     Total |       434        495        452 |      1381
```

进一步来分析,我们可以通过增加选项要求得到列的百分比(**col**)、不显示交互格频数(**nof**)和进行对独立性的卡方检验(**chi2**)。其结果揭示了一种统计性显著的关系($P = 0.001$)。想上大学的学生比例随着高中年级的升高而降低。

```
. tabulate univers year [fw = count], col nof chi2
```

```
 Expect to |
    attend |
university |      What year of school now?
         ? |        10         11         12 |     Total
-----------+---------------------------------+----------
        no |     48.39      52.53      60.62 |     53.87
       yes |     51.61      47.47      39.38 |     46.13
-----------+---------------------------------+----------
     Total |    100.00     100.00     100.00 |    100.00

          Pearson chi2(2) =  13.8967   Pr = 0.001
```

基于下述一种或多种抽样方法,调查数据往往反映了复杂的抽样设计:

非等比抽样——比如,为了对特殊子群体推断有足够案例,从而对他们进行过度抽样(oversampling)。

整群抽样——比如,首先随机选择选区,然后在抽中选区内调查所有个体。

分层抽样——比如,首先将选区分成"城镇"和"乡村"两层,然后在每一层内分别抽取选区和个体。

复杂的抽样设计要求专门的分析工具。对此,**pweights** 和 Stata 的常规分析命令并不能满足。

Stata 针对复杂调查数据的程序包括特殊表格、平均数、回归、logit、probit、tobit 和泊松回归等命令。在应用这些命令之前,用户必须首先按识别变量设置好数据,包括区分基本抽样单位(PSU)、或者群、或者层,提供有限总体修正系数和概率权数。这都可以通过 **svyset** 命令加以实现。比如:

```
. svyset precinct [pweight=invPsel], strata(urb_rur) fpc(finite)
```

对于本例中的每一观测,变量 *princinct* 的值标示了 PSU 或群。变量 *urb_rur* 的取值标示了层,*finite* 给出了有限总体修正系数,同时 *invPsel* 给定了概率权数或抽中概率的倒数。数据经 **svyset** 处理、并被保存之后,调查的分析程序就相对简单了。一般,我们需要在命令的前面加上前缀 **svy:**,如下所示:

```
svy:  mean income
```

或者,

```
svy:  regress income education experience gender
```

《调查数据参考手册》(*Survey Data Reference Manual*)囊括了 Stata 广泛的调查分析能力的详细说明和示例。对于在线指导,请键入 **help svy** 并按有关链接找到特定命令。

生成随机数据和随机样本

伪随机数函数 **uniform( )** 集中体现了 Stata 生成随机数据或对现有数据进行随机抽样的能力。《基础参考手册》(各函数)提供了关于 32 比特伪随机数发生器的技术描述。如果目前内存中读入了数据,那么以下命令可生成一个名为 *randnum* 的新变量,对于数据中的每一案例,该变量显然是从区间[0,1)内随机抽取的 16 位数值。

```
. generate randnum = uniform()
```

作为替代,我们也可以从内存创建一个随机数据集。假如我们想创建一个包含 10 个随机数的新数据,首先需要将内存中的任何其他数据清除掉(如果有价值的话,请先用

save 命令保存它们)，接下来设定新数据中想要的观测案例数。明确地设定种子数能使以后重新得到同样的"随机"结果。最后，生成我们的随机变量。

```
. clear
. set obs 10
obs was 0, now 10
. set seed 12345
. generate randnum = uniform()
. list

     +----------+
     |  randnum |
     |----------|
  1. |   .309106 |
  2. | .6852276 |
  3. | .1277815 |
  4. | .5617244 |
  5. | .3134516 |
     |----------|
  6. | .5047374 |
  7. | .7232868 |
  8. | .4176817 |
  9. | .6768828 |
 10. | .3657581 |
     +----------+
```

结合 Stata 的代数函数、统计函数和特殊函数，**uniform( )**可以模拟由不同理论分布抽取数值。如果我们想要为新变量 *newvar* 从区间 [0,428) 内、而不是从常用的区间 [0,1) 内抽取均匀分布(uniform distribution)的数值，我们就键入：

```
. generate newvar = 428 * uniform()
```

所取得的将仍然是 16 位的数值。也许，我们只想要从 1 到 428 之间(含两端)的整数，那么键入：

```
. generate newvar = 1 + trunc(428 * uniform())
```

为了模拟 1 000 次投掷一个骰子的结果，键入：

```
. clear
. set obs 1000
obs was 0, now 1000
. generate roll = 1 + trunc(6 * uniform())
. tabulate roll

        die |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |        171       17.10       17.10
          2 |        164       16.40       33.50
          3 |        150       15.00       48.50
          4 |        170       17.00       65.50
          5 |        169       16.90       82.40
          6 |        176       17.60      100.00
------------+-----------------------------------
      Total |       1000      100.00
```

从理论上，我们可以预期出现 1 点的情形占 16.67%，出现 2 点的情形占 16.67%，如此等等，但是在任何一个抽取的样本中，比如，这 1 000 次掷骰子，其观测百分比将围绕其期望值随机波动。

还可以模拟 1 000 次同时掷两个骰子的结果，键入：

```
. generate dice=2+trunc(6*uniform())+trunc(6*uniform())

. tabulate dice

       dice |      Freq.     Percent        Cum.
------------+-----------------------------------
          2 |         26        2.60        2.60
          3 |         62        6.20        8.80
          4 |         78        7.80       16.60
          5 |        120       12.00       28.60
          6 |        153       15.30       43.90
          7 |        149       14.90       58.80
          8 |        146       14.60       73.40
          9 |         96        9.60       83.00
         10 |         88        8.80       91.80
         11 |         53        5.30       97.10
         12 |         29        2.90      100.00
------------+-----------------------------------
      Total |       1000      100.00
```

我们也可以使用 _n 来生成一个人工数据。以下命令创建一个新的有5 000 个观测的数据,该数据只有一个取值从1 到5 000 的名为 *index* 的变量。

```
. set obs 5000
obs was 0, now 5000

. generate index = _n

. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       index |      5000      2500.5     1443.52          1       5000
```

也可以使用 **uniform( )** 生成服从正态(高斯)分布的变量。下述例子创建一个包含2 000 观测案例和 *z* 与 *x* 两个变量的数据,其中 *z* 来自于N(0,1)分布的总体,*x* 来自于N(500,75)分布的总体。

```
. clear

. set obs 2000
obs was 0, now 2000

. generate z = invnormal(uniform())

. generate x = 500 + 75*invnormal(uniform())
```

实际样本的平均数和标准差会略微不同于它们的理论值。

```
. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           z |      2000    .0375032    1.026784  -3.536209   4.038878
           x |      2000     503.322    75.68551   244.3384   743.1377
```

如果 z 服从正态分布,那么 $v = e^z$ 就服从对数正态分布(lognormal distribution)。根据标准正态分布的 z 可以生成一个服从对数正态分布的变量 v:

```
. generate v = exp(invnormal(uniform()))
```

要想根据N(100,15)分布生成一个服从对数正态分布的变量 w,键入命令:

```
. generate w = exp(100 + 15*invnormal(uniform()))
```

当然,取对数又将一个对数正态变量加以正态化(normalize)。

为了模拟从一个有平均数和标准差 $\mu = \sigma = 3$ 的指数分布(exponential distribution)总体中随机抽取的变量 y 的值,用

```
. generate y = -3 * ln(uniform())
```

对于平均数和标准差为其他值的情况,用其他值替代 3 即可。

假如 *X1* 服从自由度为 1 的卡方分布,它就与标准正态的平方完全一样:

```
. generate X1 = (invnormal(uniform())^2
```

根据类似的逻辑,要生成 *X2* 服从自由度为 2 的卡方分布,用:

```
. generate X2=(invnormal(uniform()))^2+(invnormal(uniform()))^2
```

其他的统计分布,包括 t 分布和 F 分布,也可以采用同样的方式进行模拟。此外,还有为 Stata 编好的程序可以生成二项(binomial)分布、泊松(Poisson)分布和逆高斯(inverse Gaussian)分布等随机样本。

尽管 **invnormal(uniform())** 经过调整可用于形成有特定相关关系的不同正态变量,更简单的办法则是使用 **drawnorm** 命令。为了生成服从 N(0,1)分布的 5 000 个观测案例,键入:

```
. clear
. drawnorm z, n(5000)
. summ

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           z |      5000   -.0005951    1.019788  -4.518918   3.923464
```

下面,我们将进一步创建三个变量。变量 *x1* 来自 N(0,1)分布总体,变量 *x2* 来自 N(100,15)分布总体,而变量 *x3* 则来自 N(500,75)分布总体。而且,我们限定这些变量之间具有以下的总体相关关系:

	x1	**x2**	**x3**
x1	1.0	0.4	-0.8
x2	0.4	1.0	0.0
x3	-0.8	0.0	1.0

创建此类数据的程序需要首先定义相关矩阵 *C*,然后在 **drawnorm** 命令中调用 *C*:

```
. mat C = (1, .4, -.8 \ .4, 1, 0 \ -.8, 0, 1)
. drawnorm x1 x2 x3, means(0,100,500) sds(1,15,75) corr(C)
. summarize x1-x3

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          x1 |      5000    .0024364     1.01648  -3.478467   3.598916
          x2 |      5000    100.1826    14.91325   46.13897   150.7634
          x3 |      5000    500.7747    76.93925   211.5596   769.6074

. correlate x1-x3
(obs=5000)

             |       x1       x2       x3
-------------+---------------------------
          x1 |   1.0000
          x2 |   0.3951   1.0000
          x3 |  -0.8134  -0.0072   1.0000
```

将样本的相关系数和平均数与前面给出的理论值加以对比。用这种方式生成的随机数据可以看作是从理论总体中抽取的样本。我们不应当期望样本的统计与理论总体的参数完全相等。(本例中,*x3* 的平均数为 500,*x1* 与 *x2* 的相关为 0.4,*x1* 与 *x3* 的相关为

-0.8,等等)

命令 **sample** 可以利用 **uniform** 的随机数发生器来抽取内存数据的随机样本。比如,为了从原始数据中提取一个 10% 的随机样本,可键入:

```
. sample 10
```

当我们加上 **in** 或 **if** 限制条件时,**sample** 只应用于那些满足条件的观测案例。比如:

```
. sample 10 if age < 26
```

将保留那些年龄小于 26 岁的观测案例的 10% 样本,同时还会保留所有年龄大于等于 26 岁的原始观测案例。

我们也可以选择某一特定规模的随机样本。为了从内存中的原始数据中随机选取 90 个观测案例,可以键入:

```
. sample 90, count
```

第 14 章中关于自助法和蒙特卡罗模拟的部分进一步提供了随机抽样和随机变量生成的例子。

编制数据管理程序

大规模的数据管理常常需要反复处理,又很容易出错,因此最好通过编制专门 Stata程序来进行处理。高级编程可能是技术性很强的工作,但是我们也可以从编写只包含一些 Stata 命令的简单程序开始,并将其存成一个 ASCII 文件。用户可以使用所喜欢的文字处理器或文本编辑器来创建 ASCII 文件,当然,这些编辑器应当能在 **File-Save As** 下的选项内提供"ASCII 文本文件"的文件类型。使用 Stata 的 Do 文件编辑器创建此类文本文件就更为简单,它可以通过点击 **Window-Do-file Editor** 或图标钮来启动。此外,还可以通过键入命令 **doedit** 或在文件已经存在的情况下键入 **doedit *filename*** 来启动 Do 文件编辑器。

比如,我们使用 Do 文件编辑器来创建一个名为 *canada.do* 的文件(其中包含了从名为 *canada.raw* 的原始数据文件读取原始数据的命令),然后为数据及其变量添加标签,压缩数据并将其存成 Stata 格式。如果我们逐步查看该例子,会发现该文件中的命令内容和前面一步一步所做的完全一样。

```
infile str30 place pop unemp mlife flife using canada.raw
label data "Canadian dataset 1"
label variable pop "Population in 1000s, 1995"
label variable unemp "% 15+ population unemployed, 1995"
label variable mlife "Male life expectancy years"
label variable flife "Female life expectancy years"
compress
save canada1, replace
```

一旦 *canada.do* 文件编写完成并加以保存,只要简单键入以下命令就可使得 Stata读取该文件并逐步执行其中的每一条命令:

```
. do canada
```

此类被称作"do 文件"的批模式程序常常保存成以 .do 作为扩展名的文件。更为精致的程序(由 do 文件或"自动的 do"文件加以定义)可以保存于内存中,并可以再调用其

他程序，这就创建出新的 Stata 命令，并为喜欢尝试各种探索的分析人员开启了可能性的世界。**Do** 文件编辑器还具有几个其他的有用功能。第 3 章描述了一种使用 do 文件来创建图表的简单方法。进一步的信息，请见《入门手册》(*Getting Started manual*)有关使用 Do 文件编辑器的说明。

Stata 通常把一条命令行的结尾视为该命令的结束。这在屏幕显示时是可行的，此时的命令行可以为任意长度，但是当我们将命令写入一个文本文件中时，这样就行不通了。这时可采用**#delimit** 命令来解决命令行宽限制的问题，该命令可以设定其他一些符号来作为一行命令结束的分隔符。在下述例子中，我们使用英文分号作为分隔符，接着键入两条很长的命令，都是直到分号出现才结束；最后再将分隔符重新设定回其常规值，即回车(cr)：

```
#delimit ;
infile str30 place pop unemp mlife flife births deaths
   marriage medinc mededuc using newcan.raw;
order place pop births deaths marriage medinc mededuc
   unemp mlife flife;
#delimit cr
```

每当显示结果占满了结果窗口(Result window)时，Stata 通常会暂停，直到我们敲任意键(或点击按钮)才会继续。为了不出现暂停，我们可以要求 Stata 持续翻屏直到输出结果全部完成。将下述命令键入命令窗口(Command window)中或作为程序的一部分：

```
. set more off
```

表示要求持续翻屏。当程序产生许多屏我们并不想看的输出结果、或者当结果被写入一个我们可以随后再查看的 log 文件时，这就变得很方便。如果键入：

```
. set more on
```

则是要求重新回到在翻屏之前等待键盘输入的常规状态。

内存管理

当我们打开(**use** 或 **File-Open**)一个数据集时，Stata 读取保存在磁盘上的文件并将其载入内存。将数据载入内存可以使分析做得更快，但是这只是在数据适合当前分配给 Stata 的内存空间时才可行。如果试图打开一个过大的数据，我们就会得到一条特定的错误信息，说“没有空间来加入更多观测”(no room to add more observations)，并且还提示了下一步的处理建议：

```
. use C:\data\gbank2.dta

(Scientific surveys off S. Newfoundland)
no room to add more observations
   An attempt was made to increase the number of observations beyond what is
   currently possible.  You have the following alternatives:
    1.  Store your variables more efficiently; see help compress.  (Think of
        Stata's data area as the area of a rectangle; Stata can trade off width
        and length.)
    2.  Drop some variables or observations; see help drop.
    3.  Increase the amount of memory allocated to the data area using the set
        memory command; see help memory.
r(901);
```

Small Stata 版本会对数据分配固定的内存量,并且这一限度不能更改。但是,Intercooled Stata 和 Stata/SE 这两个版本却是灵活可变的。Intercooled 版本默认的内存分配量为 1 兆字节,而 Stata/SE 版本则为 10 兆字节。如果我们用的是 Intercooled 版本或 Stata/SE 版本,并且计算机具有足够物理内存的话,我们可以使用 **set memory** 命令来为 Stata 设定更高的内存分配量。要给数据分配 20 兆的内存空间,可以键入:

```
. set memory 20m
Current memory allocation

                    current                                 memory usage
    settable          value     description                 (1M = 1024k)
    --------------------------------------------------------------------
    set maxvar         5000     max. variables allowed           1.733M
    set memory          20M     max. data space                 20.000M
    set matsize         400     max. RHS vars in models          1.254M
                                                            -----------
                                                                22.987M
```

要是内存中已经读入数据,请首先键入命令 **clear** 来清除它们。要想使重设的内存分配"永久化",以至于我们下次启动时还是这样分配,可以键入:

```
. set memory 20m, permanently
```

在前面给出的例子中,*gbank2.dta* 是一个 11.3 兆的数据集,这并不适合于默认的内存分配量。因此,要求 20 兆的分配量现在给了我们足够的内存空间来读入这些数据。

数据集 *gbank2.dta* 包含了 1971 年到 1993 年期间对纽芬兰大浅滩的鱼类数量进行科学调查所得到的 74 078 条观测案例。当我们描述(**describe**)这些数据时(见上面的输出),Stata 报告说有"46.09% 的空闲内存",这并不是指计算机总内存的 46%,而是指分配给 Stata 数据的 20 兆中的 46%。通常,我们应该要求比数据实际所需更大的内存空间。许多统计和数据管理操作还会占用其余的内存,其中是因为它们在工作过程中会临时性地创建一些新变量。

将内存设置(**set memory**)到超过计算机可用物理内存也是可能的。此时,Stata 会使用实际上存在于磁盘存储器上的"虚拟内存"。尽管虚拟内存允许超过硬件限制,但是计算机运行会变得极慢。如果用户总是在超出计算机限制的情况下处理数据,那就意味着您很快就会决定去购买更多内存了。

请键入 **help limits** 查看 Stata 中的限制清单,不但有关于数据集大小的限制,也有包括矩阵大小、命令长度、名称长度和命令中的变量数等其他方面的限制。其中的某些限制,用户可以自行调整。

```
Contains data from C:\data\gbank2.dta
  obs:        74078                        Spring scientific surveys NAFO
                                             3KLNOPQ, 1971-93
 vars:           44                        2 Mar 2000 21:28
 size:     11333934 (46.0% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              float  %9.0g                  original case number
rec_type        byte   %4.0g
vessel          byte   %4.0g                  Vessel
trip            int    %8.0g                  Trip number
set             int    %8.0g                  Set number
rank            int    %8.0g
assembla        str7   %7s
year            byte   %4.0g                  Year
month           byte   %4.0g                  Month
day             byte   %4.0g                  Day
set_type        byte   %8.0g       set_type   Set type
stratum         int    %8.0g                  Stratum or line fished
division        str2   %2s                    NAFO division
unit_are        str3   %3s                    Nfld. area grid map square
light           int    %8.0g                  Light conditions
wind_dir        byte   %4.0g                  Wind direction
wind_for        byte   %4.0g                  Wind force
sea             byte   %4.0g
bottom          byte   %4.0g                  Type of bottom
time_mid        int    %8.0g                  Time (midpoint)
duration        byte   %8.0g                  Duration of set
tow_dist        int    %8.0g                  Distance towed

gear_op         byte   %4.0g                  Operation of gear
depthcat        byte   %4.0g                  Category of depth
min_dept        int    %8.0g                  Depth (minumum)
max_dept        int    %8.0g                  Depth (maximum)
bot_dept        int    %8.0g                  Depth (bottom if MWT)
temp_sur        int    %8.0g                  Temperature (surface)
tempcat         byte   %8.0g                  Category of temperature
temp_fs_        int    %8.0g                  Temperature (fishing depth)
lat             float  %9.0g                  Latitude (decimal)
long            float  %9.0g                  Longitude (decimal)
pos_meth        byte   %4.0g
gear            int    %8.0g                  Gear
total           byte   %9.0g
species         int    %8.0g                  Species
number          long   %9.0g                  Number of individual fish
weight          double %9.0g                  Catch weight in kilograms
latin           str31  %31s                   Species -- Latin name
common          str27  %27s                   Species -- common name
surtemp         float  %9.0g                  Surface temperature degrees C
fishtemp        float  %9.0g                  Fishing depth temperature C
depth           int    %9.0g                  Mean trawl depth in meters
ispecies        byte   %9.0g                  Indicator species
-------------------------------------------------------------------------------
Sorted by:  id
```

3 制 图

作为对 Stata 分析结果含义以及综合其他分析的一种展示，图形出现在本书的每一章。确实，图形一直是 Stata 的强项，也是许多用户选择 Stata 而舍弃其他软件包的理由。从 Stata 的第 1 版到第 7 版，**graph** 命令逐渐发展。但是，Stata 第 8 版在制图方面取得了重大进步。**graph** 经过了基础性的重新设计，增强了形成精致的、符合发表质量要求的分析图形的能力。输出图形的外观和选择也得以大大改进。使用新的 **graph** 命令语法和默认设定，或者代之以通过新菜单的方式，很容易就能画出具有吸引力的（可供发表的）基本图形。那些并不满足于基本图形而在制图方面要求更高的用户将会发现他们可以确实得到一系列工具和选项上的支持，这些功能在 500 页的《制图参考手册》（*Graphics Reference Manual*）中作了描述。

在本章简短的篇幅内，我们将采用制图示例方式，而不是命令语法分析方式，来介绍从基础制图到创造性制图的广泛内容（请见《制图参考手册》或键入 **help graph** 查看全部的命令内容）。我们从说明七种基本图形开始。

histogram	直方图
graph twoway	双变量的散点图（scatterplot）、曲线标绘图（line plot）和许多其他图形
graph matrix	散点图矩阵（scatterplot matrix）
graph box	箱线图（box plot）
graph pie	饼图（pie plot）
graph bar	条形图（bar plot）
graph dot	点图（dot plot）

这些基本类型中的每一种都包含许多选项。对于功能强大的 **twoway** 图型而言尤其如此。

为了查看变量分布的详细情况，还有诸如对称图（symmetry plot）、分位数标绘图（quantile plot）和分位—正态图（quantile-normal plot）等更专门的图形。本章包括了这些图形以及工业质量控制图的一些例子。请键入 **help graph_other** 查看更多细节。

最后，本章结束时将讲解一些能够表达丰富数据，并能用于发表的完备图形的有用技术。此类技术包括为图形添加文本、叠并多个二维图、保存并取回图形加以格式编辑，以及将多个图形合而为一。随着发展，制图命令变得更为复杂，因此写出（do 文件形式的）简单的批程序（batch programs）可以便利重新使用这些图形。制图选项的全部内容远远超出了本书覆盖的范围，但是本章结束时将会用例子指点一些制图的可能性，随

后各章还会提供更多的例子。

制图(**Graphics**)菜单使得通过点击方式也能完成大部分的制图工作。

<u>对Stata长期用户的一点提示</u>:Stata第8版和第9版的制图能力使得那些更早版本相形见绌。对于习惯于旧版本Stata的分析人员而言,有许多新材料要学习。菜单方式提供了一条入门捷径,同时,与原有制图命令一样,新的制图命令遵循一致的逻辑,通过实习就能搞清楚。幸运的是,这种变化并不是突然的。如果需要,第7版的制图命令仍然可用。它们已被移到命令 **graph7** 之下。比如,旧版本的散点图以前是采用以下命令:

. **graph** ***income education***

这个命令在更新版本的Stata中无法运行。作为替代,命令

. **graph7** ***income education***

将再现熟悉的旧版图形。**graph7** 的选项和旧版 **graph** 的选项类似。要想查看同一散点图的升级版,请键入新的制图命令:

. **graph twoway scatter** ***income education***

新命令的更多例子将在随后各节给出,它们可以说明重新设计的图形能力发生了哪些变化(以及哪些与以前一样)。

命令示范

. **histogram** ***y***, **frequency**

画出变量y的直方图,以纵轴(vertical axis)显示频数。

. **histogram** ***y***, **start(0) width(10) norm fraction**

*x*轴以0处为起点,画出变量*y*的直方图,条宽度为10。根据样本平均数和标准差添加正态曲线,并在纵轴上显示出小数形式(fraction)的数据频率。

. **histogram** ***y***, **by(*x*, total) fraction**

在一幅图中,对*x*的每个取值画出*y*的各个直方图,同时画出样本整体的"总"直方图。

. **kdensity *x*, generate(*xpoints xdensity*) width(20) biweight**

计算并画出*x*分布的内核密度(kernel density)估计值。创建出两个新变量:*xpoints*为要估计密度的*x*的各点取值;*xdensity*为相应的密度估计值。**width(20)**以变量*x*的单位来指定内核的半宽(halfwidth of the kernel)。(如果 **width( )** 未被指定,默认值会遵循一个简单公式达到"最优")本例中的 **biweight** 选项是为了调用双加权内核(biweight kernel),而不采用默认的 **epanechnikov** 内核函数。

. **graph twoway scatter *y x***

显示*y*对*x*的基本双变量散点图。

. **graph twoway lfit *y x* || scatter *y x***

通过叠并两幅 **twoway** 图形将*y*对*x*的线性回归加以图形化:即回归(线性拟合或 **lfit**)线图和*y*对*x*的散点图。如想要给该回归线添加95%置信区间带,可用 **lfitci** 取代 **lfit**。

. **graph twoway scatter *y x*,xlabel(0(10)100) ylabel(-3(1)6,horizontal)**

建构 y 对 x 的散点图,并在 x 轴的 0,10,…,100 处加标签、在 y 轴的 -3,-2,…,6 处加标签,并且标签为水平放置而不是垂直放置(为默认状态)。

```
. graph twoway scatter y x, mlabel(country)
```

建构 y 对 x 的散点图,并且数据点标注变量 *country* 的相应取值。

```
. graph twoway scatter y x1, by(x2)
```

在一幅图中,对 x2 的每一取值画出 y 对 x1 的散点图。

```
. graph twoway scatter y x1[fweight=population], msymbol(Oh)
```

画出 y 对 x1 的散点图。标记符号为中空的圆圈(Oh),其大小与频数权数变量 *population* 成比例。

```
. graph twoway connected y time
```

y 对 *time* 的基本时间标绘图。显示的数据点由线段连接起来。要想添加线段但不要数据点标志(marker),就用 **line** 来代替 **connected**:

```
. graph twoway line y time
```

```
. graph twoway line y1 y2 time
```

画出具有相同量度的两个 y 变量对名为 *time* 的 x 变量的时间标绘图(本例中为曲线图)。

```
. graph twoway line y1 time, yaxis(1)||line y2 time,yaxis(2)
```

画出具有不同量度的两个变量的时间曲线,并将它们叠并在同一曲线标绘图内。**yaxis**(1)指定左边的 y 轴按 y1 设置量度,而 **yaxis**(2)指定右边的 y 轴按 y2 设置量度。

```
. graph matrix x1 x2 x3 x4 y
```

建构一个散点图矩阵,显示列出变量之间所有可能的两两交互散点图。

```
. graph box y1 y2 y3
```

建构变量 y1、y2 和 y3 的箱线图。

```
. graph box y, over(x) yline(.22)
```

对 x 的每一取值建构 y 的箱线图,同时在 y=0.22 处画一条水平线。

```
. graph pie a b c, pie
```

画一个饼图[1],其中的每块表明了变量 a、b 和 c 的相对量。这些变量必须具有相似的单位。

```
. graph bar (sum) a b c
```

以条形图中并排的条显示变量 a、b 和 c 各自的合计。要想得到平均数而不是合计,键入 **graph bar (mean) a b c**。其他选项还包括以条长度来表示中位数、百分位数或者每一变量的计数。

```
. graph bar (mean) a, over(x)
```

画出在变量 x 每一取值处显示变量 a 的平均数的条形图。

```
. graph bar (asis) a b c, over(x) stack
```

---

[1]【译注:选项 **pie** 后必须加括号及有关定义,如 **pie**(#,……)。】

画出变量 *a*、*b* 和 *c* 的条形图,图中变量 *a*、*b* 和 *c* 的值(照原样)在变量 *x* 的每一取值处层叠起来。

. graph dot (median) *y*, over(*x*)

画出一个点图,沿着水平刻度在 *x* 每一取值水平所对应的 *y* 的中位数处打点。其他选项包括平均数、百分位数或者每个变量的计数。

. qnorm *y*

画出一幅分位—正态标绘图(正态概率图),显示相应的正态分布的百分位数处的 *y* 的百分位数。

. rchart *x1 x2 x3 x4 x5*, connect(l)

建构一幅质量控制的 R 图,图中画出了变量 *x1* 至 *x5* 的取值范围。

图形的选项,比如,那些控制标题、标签和坐标轴上的记号标志等,在不同图形类别之间都是通用的,只要其有意义。而且,Stata 图形命令背后的逻辑在不同图形类型之间也是一致的。在将基本要件组合成图的过程中,这些共同的原理就是取得画图流畅性的关键。

直方图

直方图自身的命令 **histogram**,用于显示测量变量分布状况,是取得直方图最简单的方法。

比如,我们回到 *states.dta*,该数据包含美国 50 个州加哥伦比亚特区的一些环境和教育方面的指标(数据来源:the League of Conservation Voters, 1991; National Center for Education Statistics, 1992,1993;World Resources Institute, 1993)。

```
. use states
(U.S. states data 1990-91)

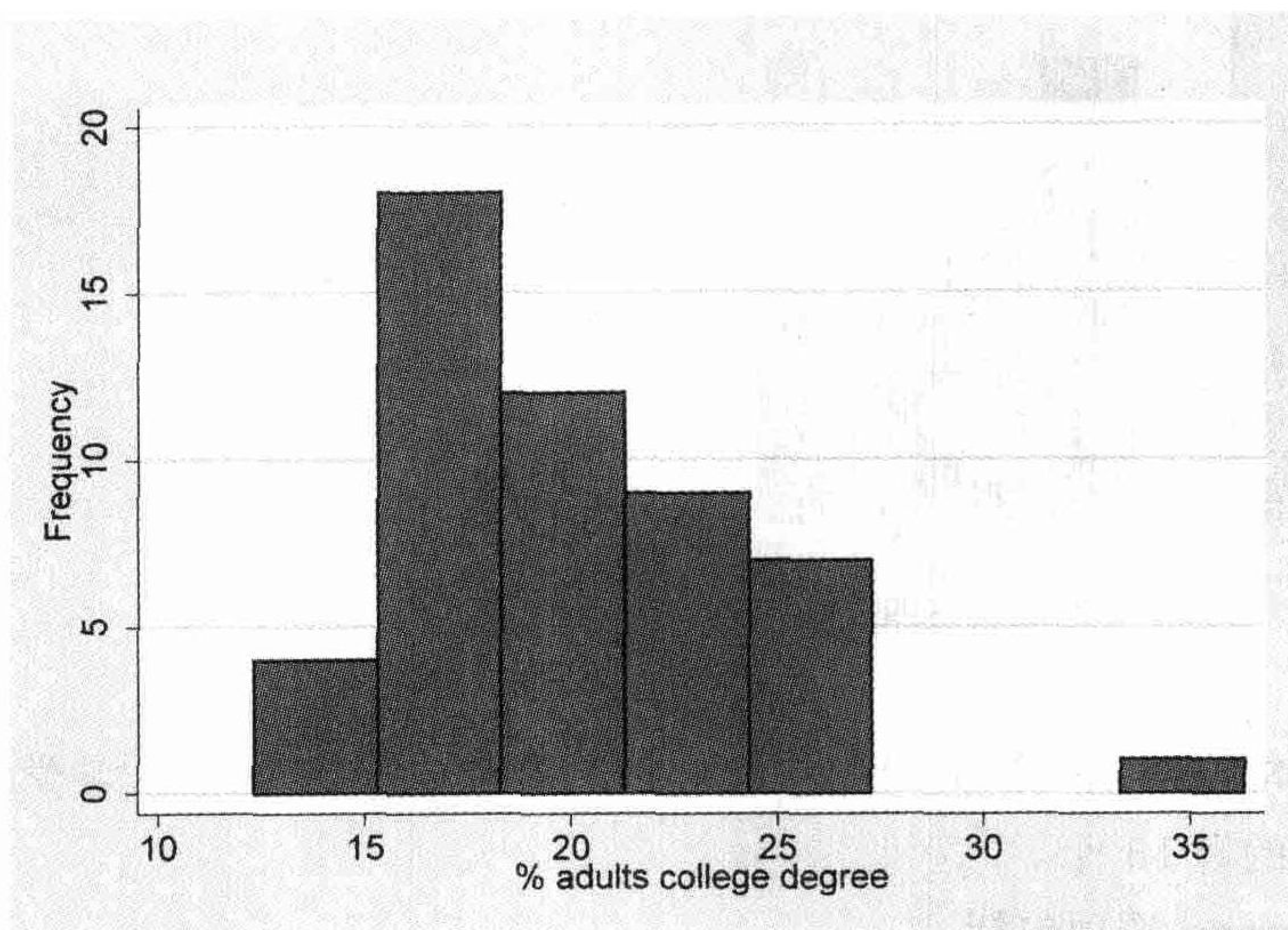
. describe
Contains data from c:\data\states.dta
  obs:            51                          U.S. states data 1990-91
 vars:            21                          4 Jul 2005 12:07
 size:          4080 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
state           str20  %20s                   State
region          byte   %9.0g       region     Geographical region
pop             float  %9.0g                  1990 population
area            float  %9.0g                  Land area, square miles
density         float  %7.2f                  People per square mile
metro           float  %5.1f                  Metropolitan area population, %
waste           float  %5.2f                  Per capita solid waste, tons
energy          int    %8.0g                  Per capita energy consumed, Btu
miles           int    %8.0g                  Per capita miles/year, 1000
toxic           float  %5.2f                  Per capita toxics released, lbs
green           float  %5.2f                  Per capita greenhouse gas, tons
house           byte   %8.0g                  House '91 environ. voting, %
senate          byte   %8.0g                  Senate '91 environ. voting, %
csat            int    %9.0g                  Mean composite SAT score
vsat            int    %8.0g                  Mean verbal SAT score

msat            int    %8.0g                  Mean math SAT score
percent         byte   %9.0g                  % HS graduates taking SAT
expense         int    %9.0g                  Per pupil expenditures prim&sec
income          long   %10.0g                 Median household income, $1000
high            float  %9.0g                  % adults HS diploma
college         float  %9.0g                  % adults college degree
-------------------------------------------------------------------------------
Sorted by:  state
```

图 3.1 显示了 college 的简单直方图,它描述了各州 25 岁以上人口中具有学士及以上学位的人口比例的分布。它可由以下命令得到:

```
. histogram college, frequency title("Figure 3.1")
```

图 3.1



在 **Prefs-Graph Preferences** 菜单中,我们有数种图形默认颜色和底色的预设方案可以选择。用户也可定义自己的方案。本书中的例子应用 **s2 mono**(单色,**monochrome**)方案,该方案下每一图形周边会有阴影边距(**shaded margins**)。**s1 mono** 方案就不会有此类的边距。尝试不同的单色和色彩方案有助于决定哪种方案特别适合某一特定目的。本章后面将会说明,在一种方案下制成并保存下来的图形随后可以取回并采用另一种方案后再次保存。

图形命令的选项可按任意顺序排列在该命令的英文逗号之后。图 3.1 示范了两个选项:要求纵轴上显示频数(而不是默认的密度);要求将标题"Figure 3.1"置于图形上方。一旦图形显示在屏幕上,就可以通过菜单选项方便地完成图形的打印、保存或者将其剪切并粘贴到其他文字处理器等程序中去。

图 3.1 表明该分布呈正偏态,众数(mode)略高于 15,存在一个取值 35 的特异值。很难更具体地去描述该图,因为图中的直方条和 x 轴的刻度并不对应。图 3.2 包含了(基于一些迅速实验所找出的恰当数值)数项改进:

1.对 x 轴添加数值标签,取值从 12 到 34,间距为 2;

2.对 y 轴添加数值标签,取值从 0 到 12,间距为 2;

3.显示 y 轴上的刻度,从 1 到 13,间距为 2;

4.直方图的第一个直方条从 12 开始;

5.每一个直方条的宽度为 2。

```
. histogram college, frequency title("Figure 3.2") xlabel(12(2)34)
    ylabel(0(2)12) ytick(1(2)13) start(12) width(2)
```

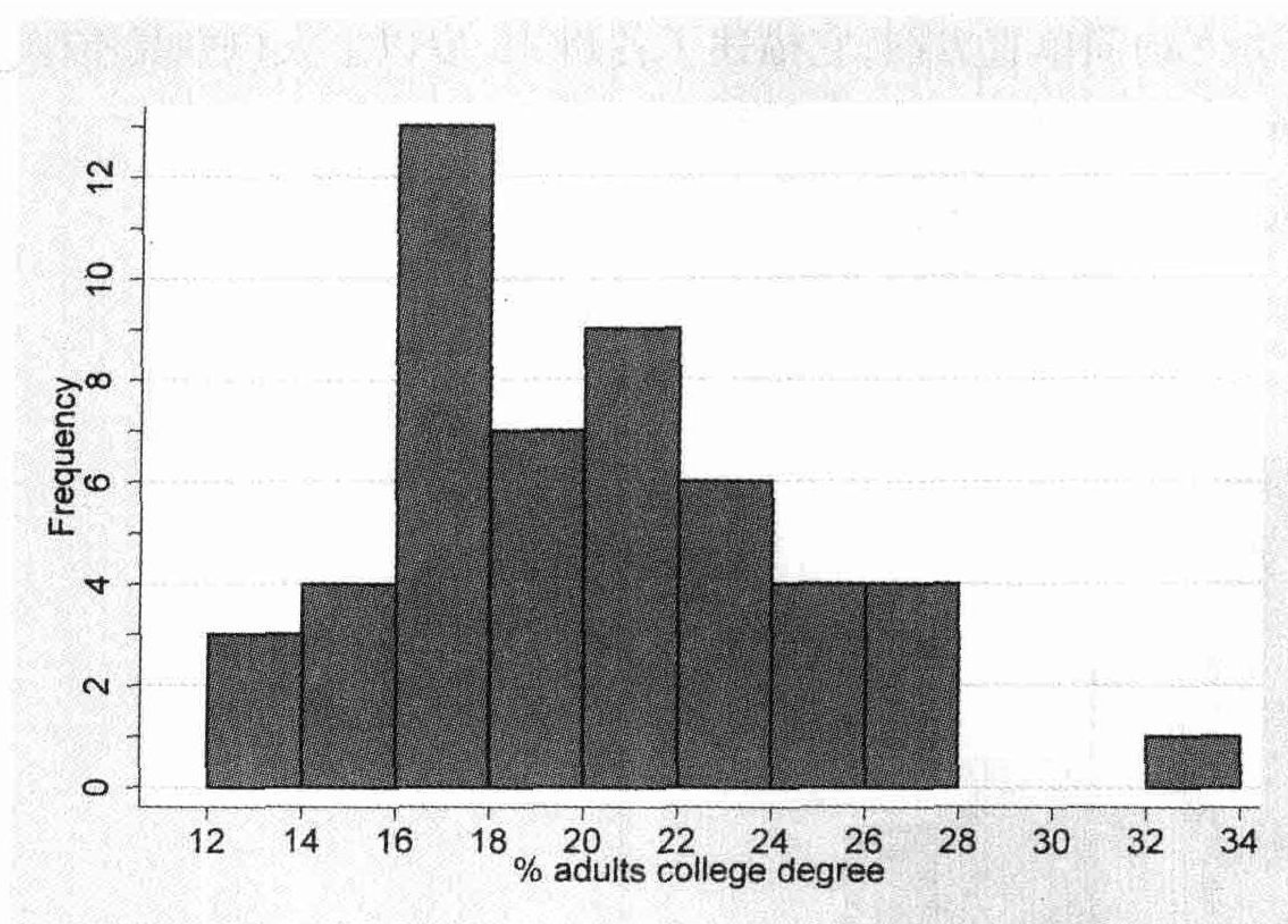


图 3.2

图 3.2 有助于我们更具体地描述该分布。比如,我们现在看到,各州中具有大学学历的百分比在 16 到 18 之间的有 13 个。

其他用的 **histogram** 选项包括:

bin(#) 图中画出的直方条为 # 个。我们要么指定 **bin(#)**,要么如图 3.2 中那样定义起始值 **start(#)** 和直方条宽度 **width(#)**,但两者不能同时存在

percent 在纵轴上显示百分比。因此,y 轴标签 **ylabel** 和 y 轴刻度 **ytick** 指的也是百分比数值。图 3.2 中还示范另一种可能选择,即频数 **frequency**。我们也可以要求显示数据的频率 **fraction**。默认状态下的直方图是显示密度 **density**,此时直方条按比例绘制,因此所有直方条的面积之和等于 1

gap(#) 要求在直方条之间留出间隙。# 是相对的,取值区间为 0 ≤ # < 100;通过试验来找到一个合适的值

addlabels 要求标注直方条的高度。另一个选项 **addlabopts** 可以改变标签的外观

discrete 定义离散数据,要求 x 的每个取值对应着一个直方条

norm 基于样本平均数和标准差,在直方图上添加一条正态曲线

kdensity 在直方图上添加内核密度(kernal-density)估计值。选项 **kdenopts** 可以改变密度的计算方法;有关细节请见 **help kdensity**

对直方图或大多数其他图形,我们也可以不理会默认设定而对横轴和纵轴制定我们自己的标题。选项 **ytitle** 控制 y 轴的标题,而 **xtitle** 控制 x 轴的标题。图 3.3 示范了这些标题,并且采用了一些其他的直方图选项。请注意在基本图形(图 3.1)发展到更复杂图形(图 3.3)过程中的新增元素。这就是 Stata 图形建构的通常模式:从简单的图形开始,然后尝试性地在原有命令基础上增加选项,在 **Review** 窗口也能选用这些选项,直到最后得到一幅能最清楚地表达研究发现的图形。图 3.3 实际上过于复杂,画在这里仅为了示范多种选项。

```
. histogram college, frequency title("Figure 3.3") ylabel(0(2)12)
    ytick(1(2)13) xlabel(12(2)34) start(12) width(2) addlabel
    norm gap(15)
```

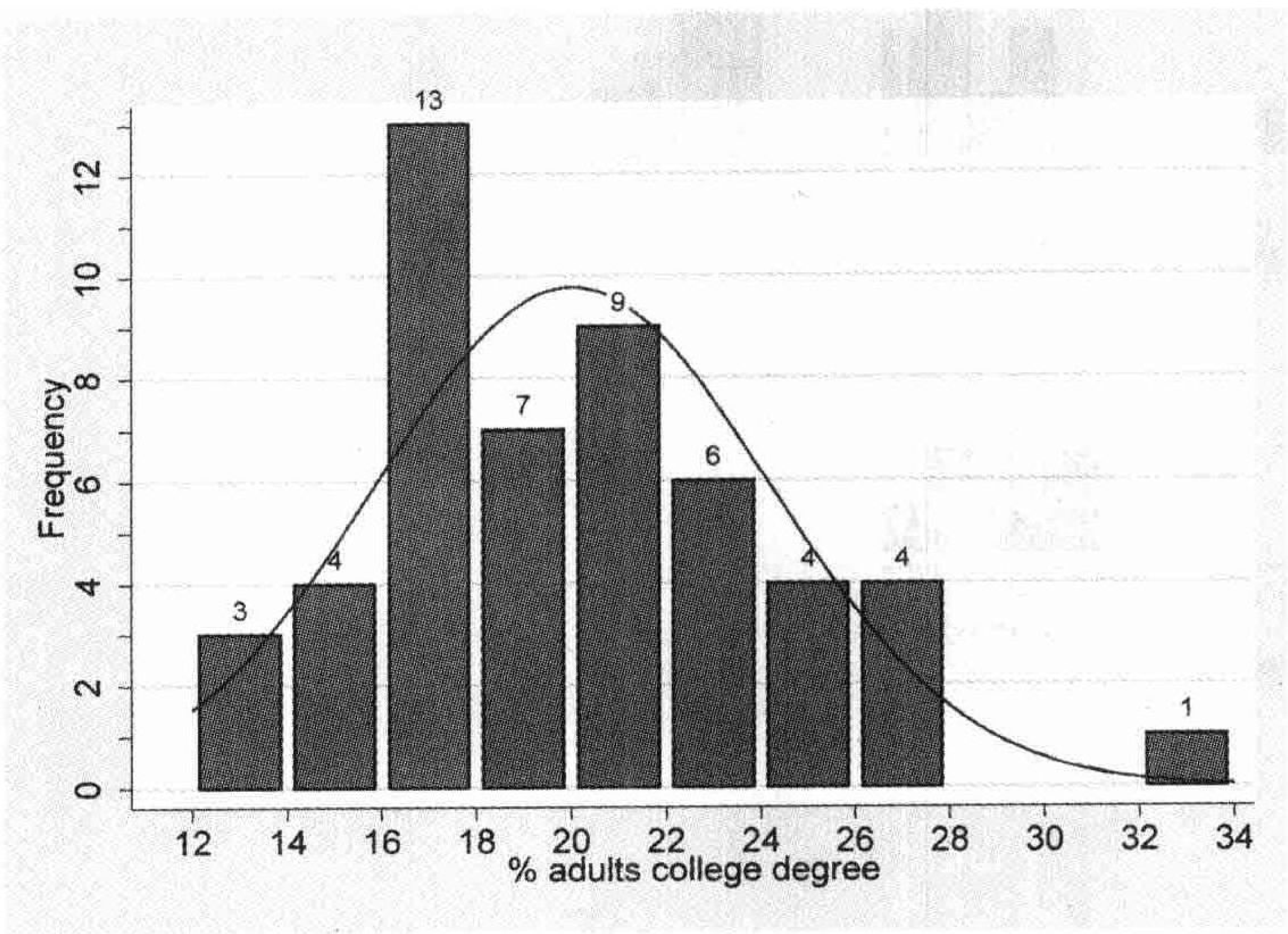


图 3.3

假如我们想看看 *college* 的分布是如何随着 *region* 而变化的。**by** 选项可以得到分别对应 *region* 每个取值的直方图。其他选项和画单个直方图时的作用相同。图 3.4展示了一个例子,其中我们要求在纵轴上显示百分比,并将数据分成 8 个直方条。

```
. histogram college, by(region) percent bin(8)
```

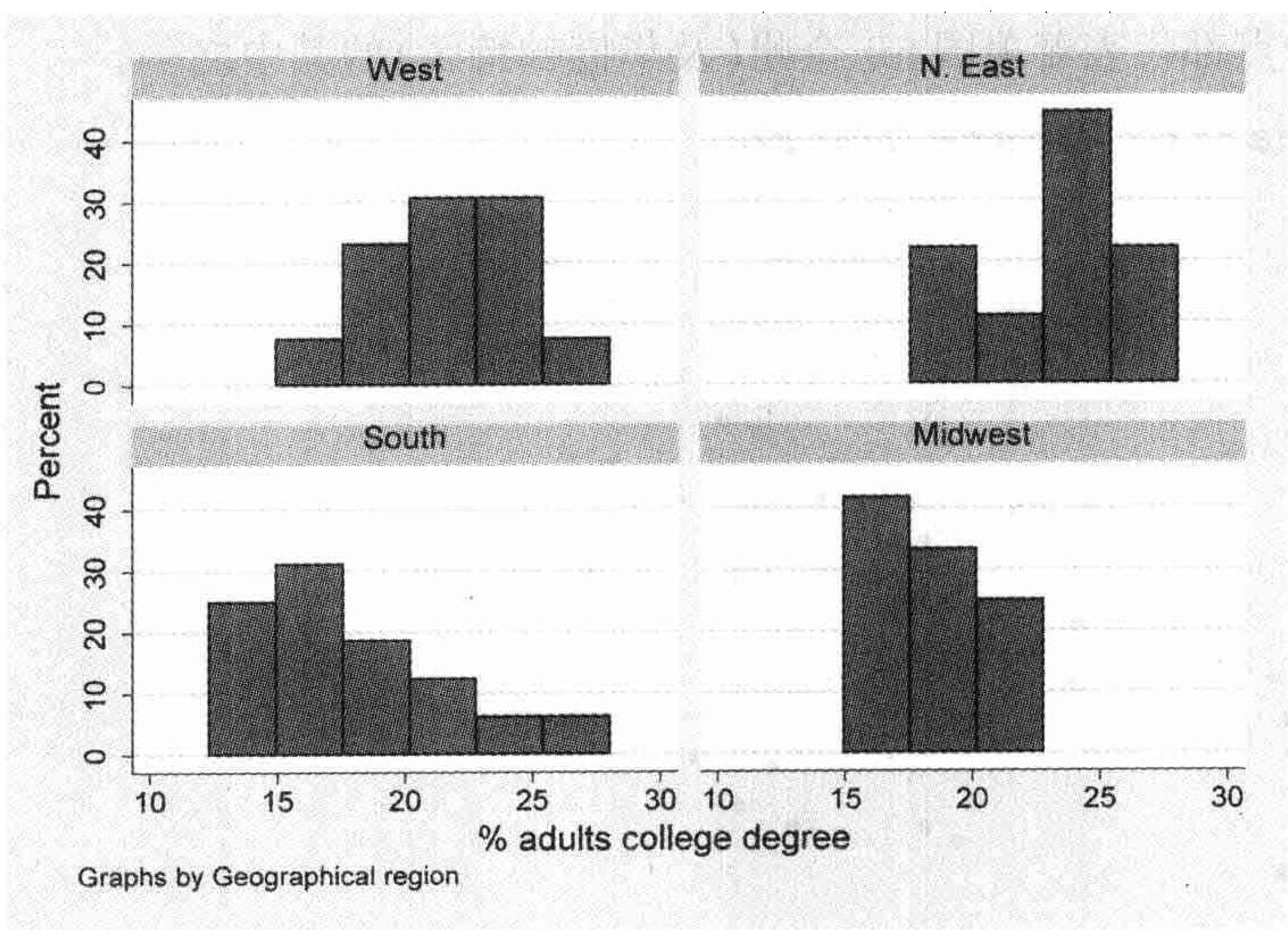


图 3.4

以下图 3.5 包括了一套四个地区的类似图形,同时加上了第五个图展示所有地区合计的分布。

```
. histogram college, percent bin(8) by(region, total)
```

坐标轴的标签、刻度、标题以及 **by(varname)** 或 **by(varname,total)**选项在其他 Stata 制图命令也起类似作用,我们在下一节就要讲到。

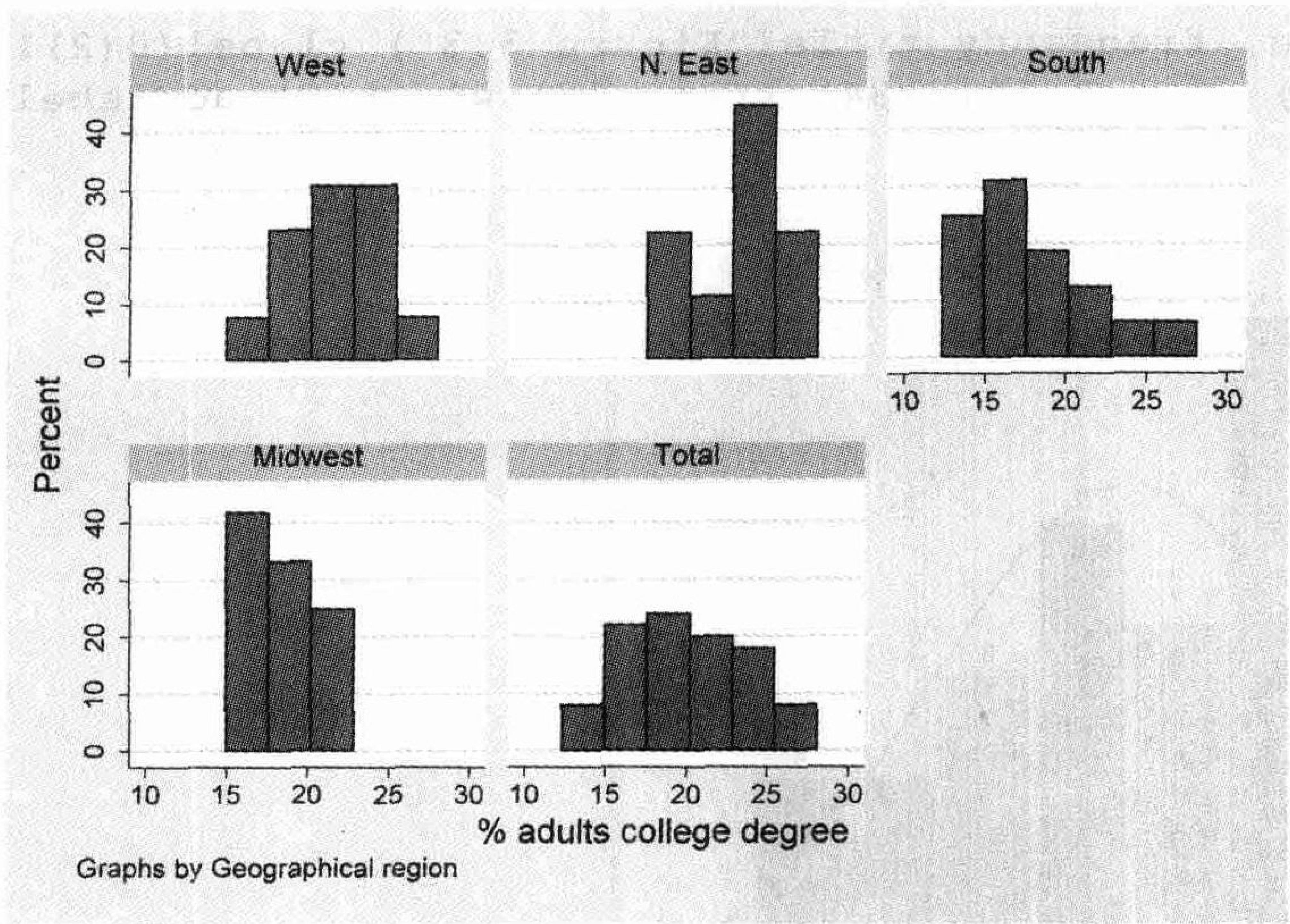


图 3.5

散点图

基本的散点图(scatterplot)可以通过一般形式的命令得到:

```
. graph twoway scatter y x
```

这里 *y* 是纵轴变量,*x* 是横轴变量。比如,还使用 *states.dta* 这一数据,我们可以作出 *waste*(人均固体废弃物)对 *metro*(大都市地区的人口比例)的散点图,结果显示在图 3.6。图 3.6 中的每一个点都代表着美国 50 个州(及华盛顿特区)的其中之一。

```
. graph twoway scatter waste metro
```

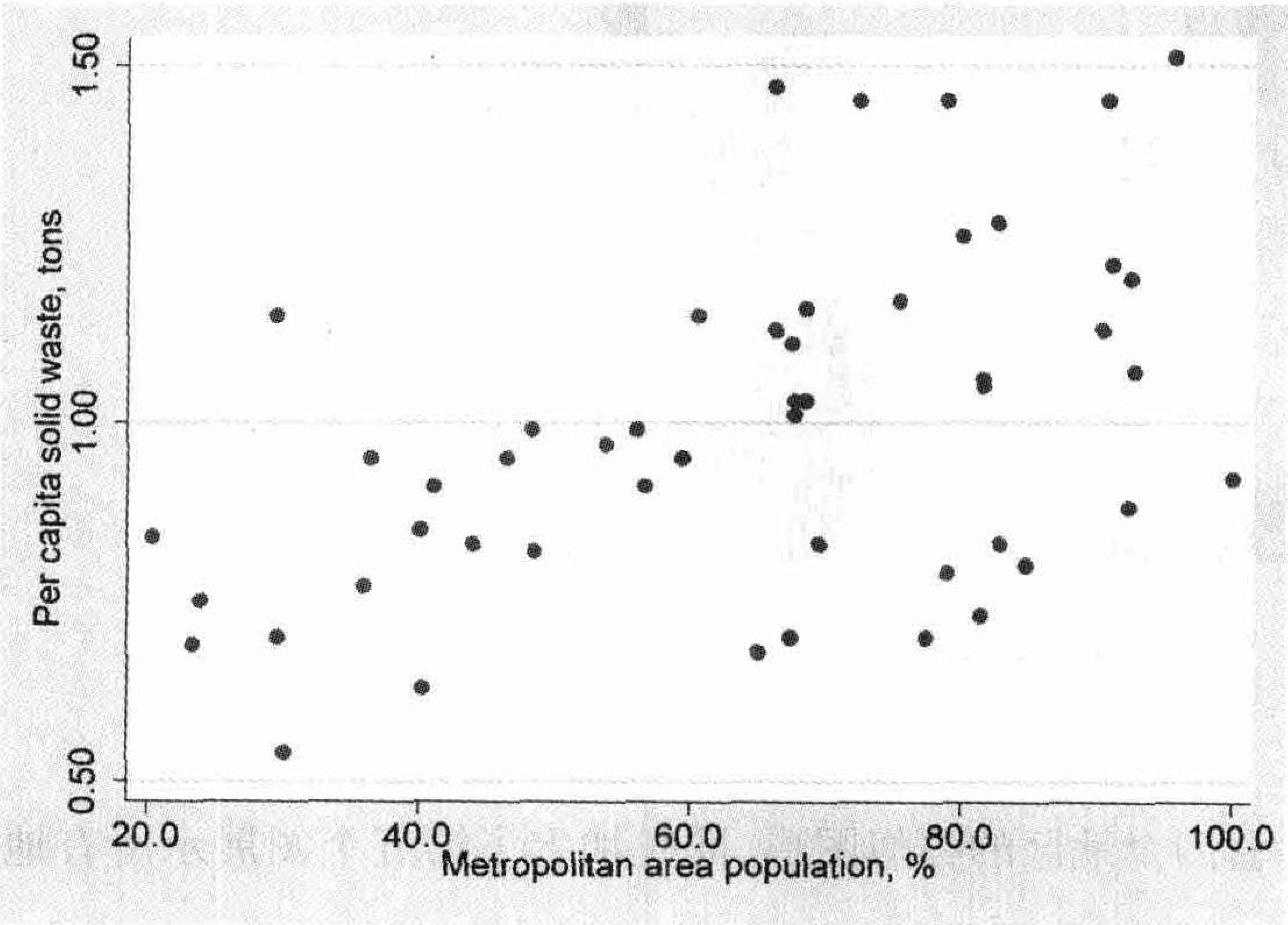


图 3.6

和直方图一样,我们也可以用 **xlabel**、**xtick**、**xtitle** 等来控制坐标轴标签、刻度或标题。散点图也允许对散点标志(marker)的形状、颜色、大小和其他属性进行控制。图 3.6 使用了实心圆圈这一默认的标志。如果我们加入选项 **msymbol(circle)** 或者简写成 **msymbol(O)** 的形式的话,将得到同样的效果。**msymbol(diamond)** 或 **msymbol(D)** 将形成一幅以菱形标志散点的图,等等。下表列出了可选的形状:

msymbol ( )	缩 写	描 述
circle	O	圆圈,实心
diamond	D	菱形,实心
triangle	T	三角,实心
square	S	方形,实心
plus	+	加号
x	X	大写字母 X
smcircle	o	小圆圈,实心
smdiamond	d	小菱形,实心
smsquare	s	小方形,实心
smtriangle	t	小三角,实心
smplus	smplus	小加号
smx	x	小写字母 x
circle_hollow	oh	圆圈,空心
diamond_hollow	Dh	菱形,空心
triangle_hollow	Th	三角,空心
square_hollow	Sh	方形,空心
smcircle_hollow	oh	小圆圈,空心
smdiamond_hollow	dh	小菱形,空心
smtriangle_hollow	th	小三角,空心
smsquare_hollow	sh	小方形,空心
point	p	很小的点
none	i	不可见

mcolor 选项控制标志的颜色。比如,命令

```
. graph twoway scatter waste metro, msymbol(S) mcolor(purple)
```

将产生一个散点图,图中的散点标志为紫色大方形。请键入 **help colorstyle** 查询可选用的颜色的清单。

散点图还能做一件有趣的事,就是可以使标志的大小(即面积)与第三个变量的取值成比例,从而赋予数据点不同的可见"权数"。比如,我们可以重画 *waste* 对 *metro* 的散点图,但使其标志的大小反映每个州的人口数(*pop*)。如图 3.7 所示,使用选项 **fweight** [ ](即频数权数)可以做到这点。空心圆圈的标志 **msymbol(Oh)** 则提供了一种更合适的形状。

频数权数对于其他一些图形类型也有用。加权可能是一个令人迷惑的复杂问题,因为"权数"来自不同类型,并且在不同情况下具有不同的含义。有关 Stata 中加权的概况,请键入 **help weight** 查询。

```
. graph twoway scatter waste metro [fweight = pop], msymbol(Oh)
```

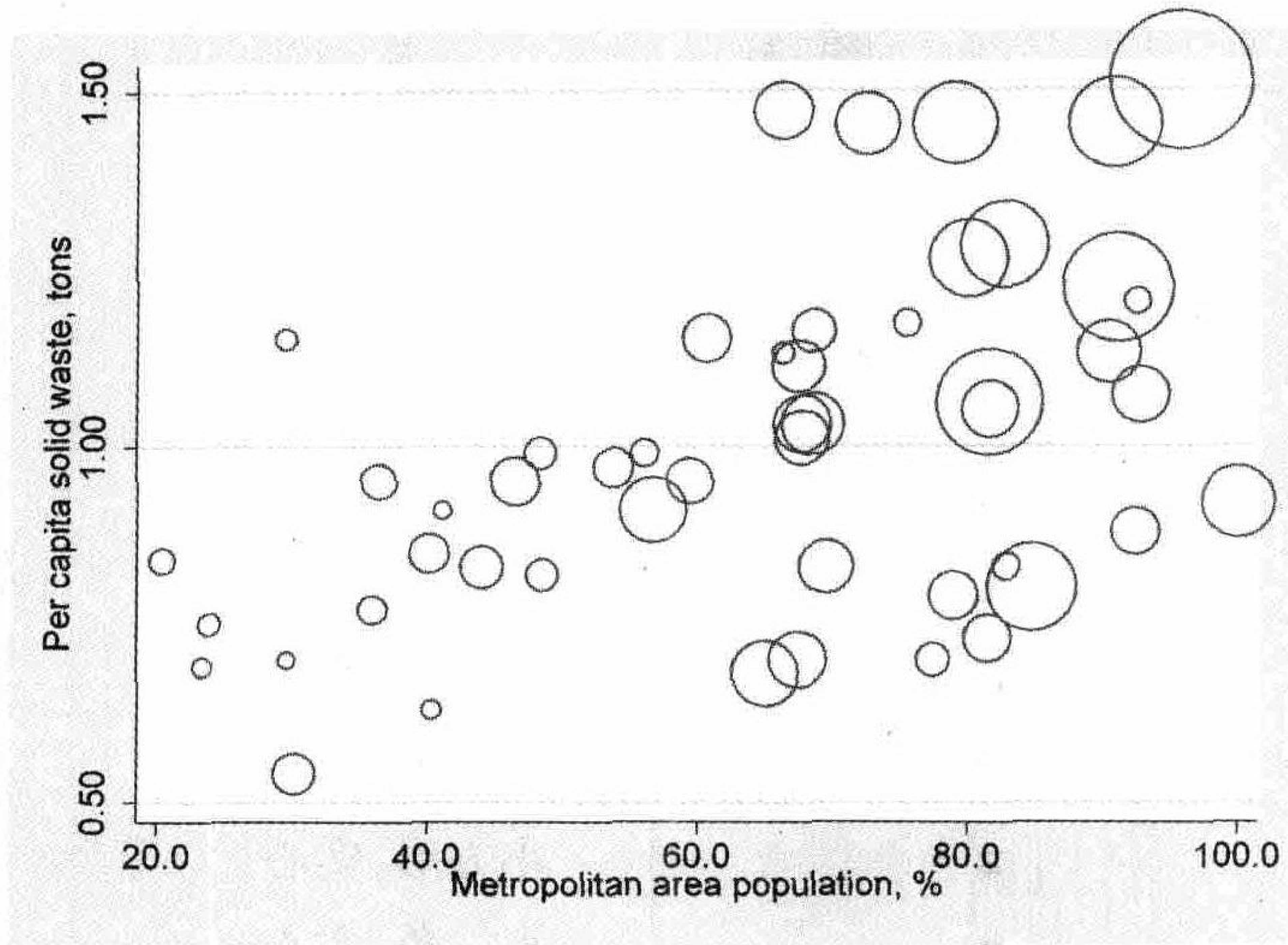


图3.7

Stata 第9版中新的密度分布葵花图(density-distribution sunflower plot)提供了一种对高密度数据画散点图的替代选择。大体上,它们类似于散点图,但其中的一些个别数据点被葵花状记号所代替以表明该处的观测案例不只一个。图3.8展示了图3.6的葵花图版,图中一些花一样的记号(有四个"花瓣"的那些)代表了该处有最多为4个州级数据点。在 **sunflower** 命令后面打印出来的表提供了有关每一朵花代表多少条观测案例的答案。花瓣(petal)的数目和花的灰度对应着数据的密度。

```
. sunflower waste metro, addplot(lfit waste metro)

Bin width            =    11.3714
Bin height           =    .286522
Bin aspect ratio     =   .0218209
Max obs in a bin     =          4
Light                =          3
Dark                 =         13
X-center             =      67.55
Y-center             =        .96
Petal weight         =          1

------------------------------------------------------------------
     flower      petal      No. of      No. of   estimated    actual
       type     weight      petals     flowers        obs.      obs.
------------------------------------------------------------------
       none                                             23        23
      light          1           3           5          15        15
      light          1           4           3          12        12
------------------------------------------------------------------
                                                        50        50
```

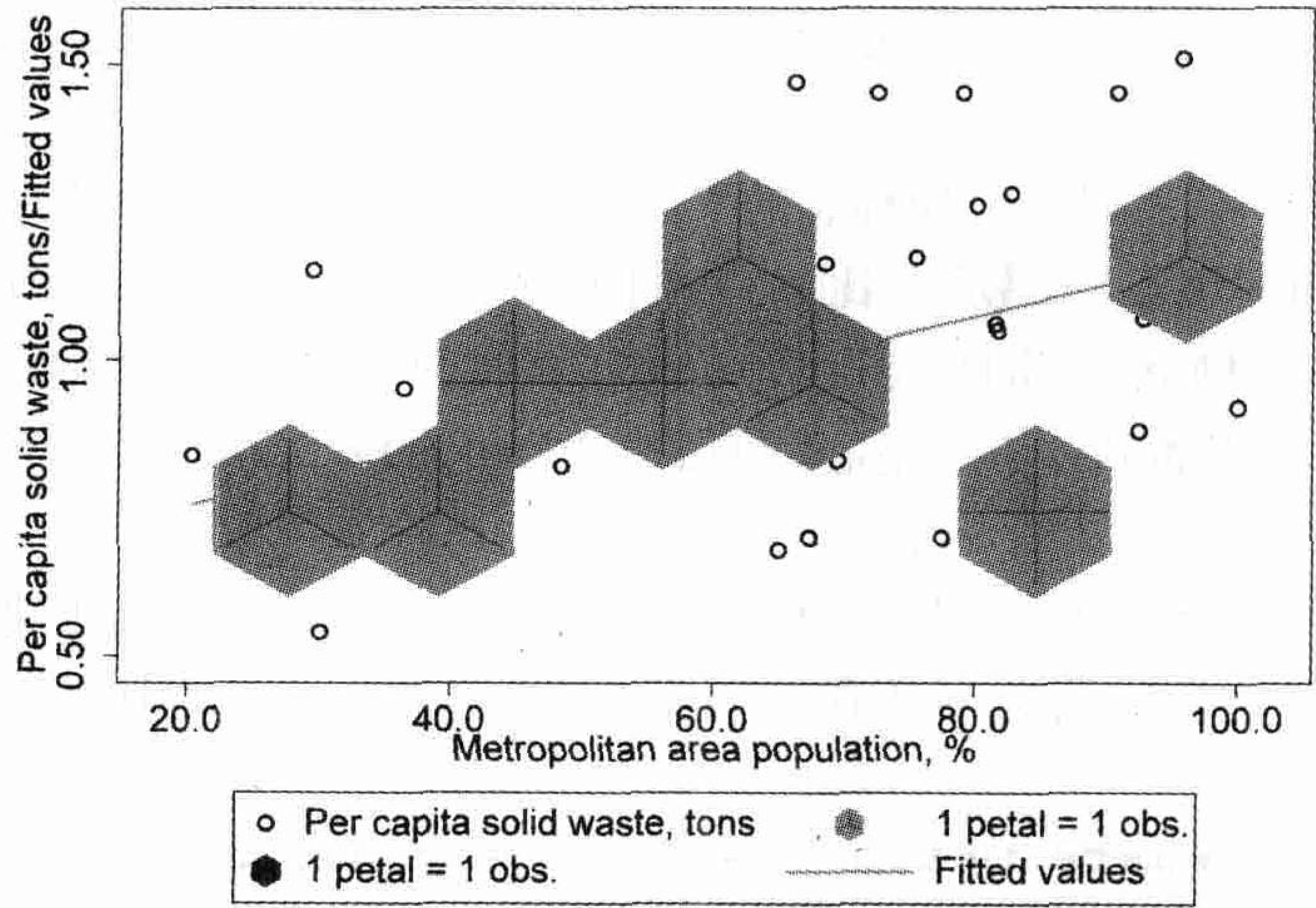


图3.8

葵花图对于大型数据或者当许多观测案例画在类似(或相同)坐标处时尤其有用。图 3.8 中的例子还添加了一条回归线,本质上是一幅 **twoway lfit** 图,只是已通过指定选项 **addplot(lfit *waste metro*)** 叠并或添加到葵花图上。

常规散点图中的标志能够根据标签进行辨别。比如,我们可能想对图 3.6 散点图中的散点加上州名。但是,50 个州的名称将会把该图变得看上去很混乱。如果只对西部这个区域似乎就更可行一些。用 **if** 选择条件来实现这一点,得到了以下图 3.9 所示的结果。

```
. graph twoway scatter waste metro if region==1, mlabel(state)
```

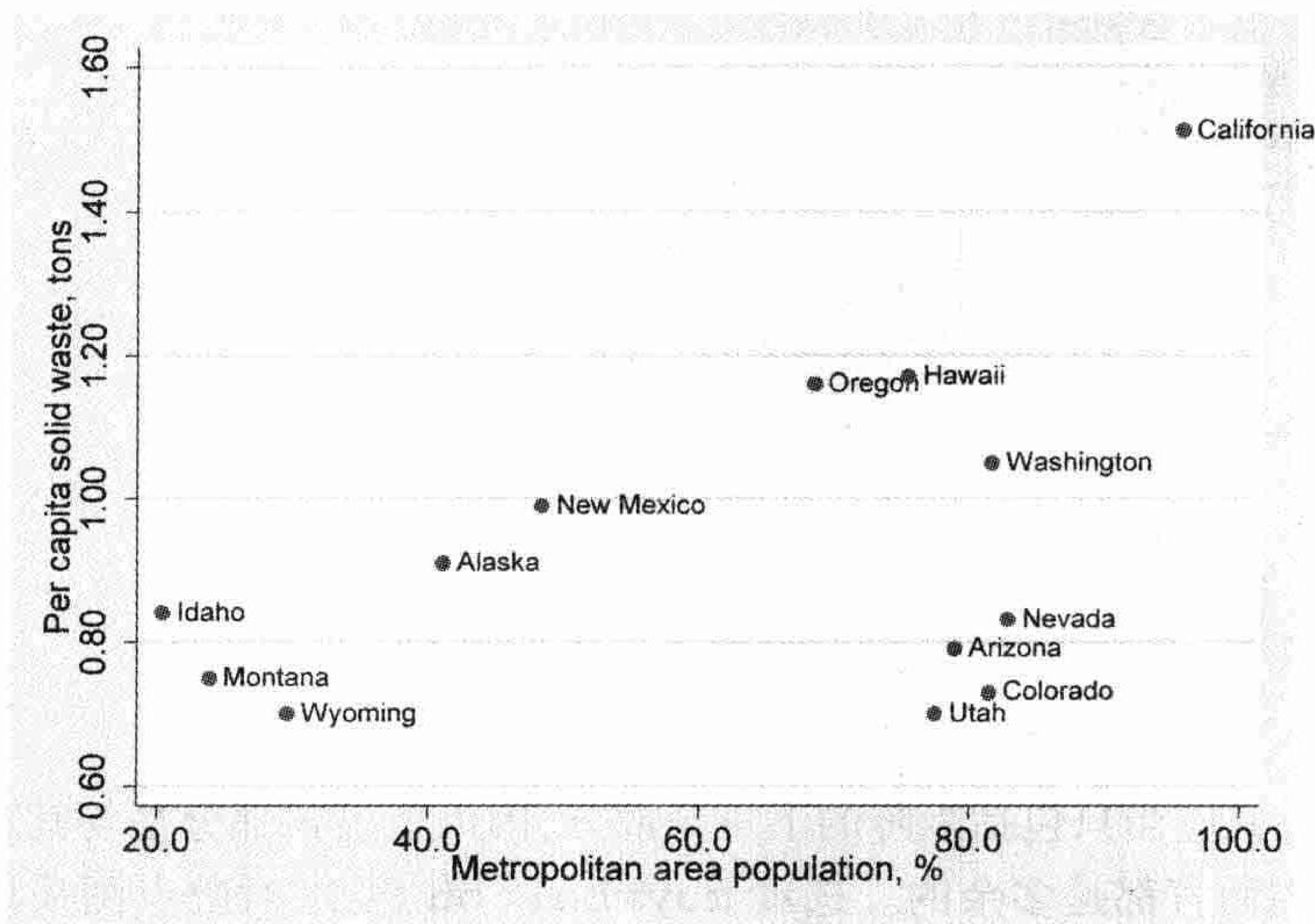


图 3.9

下面的图 3.10 显示了分别对每个区域的 *waste* 对 *metro* 的散点图。这两个变量之间的关系在南部(South)和中西部(Midwest)明显表现得比在西部(West)和东北部(N.East)要更陡,我们将在后面确认这一印象。本例中的 **ylabel** 和 **xlabel** 选项限定了 y 轴和 x 轴的标签采用不带小数的三位数(最多)的固定显示格式,这使它们在很小的分图中更易读。

```
. graph twoway scatter waste metro, by(region)
      ylabel(, format(%3.0f)) xlabel(, format(%3.0f))
```

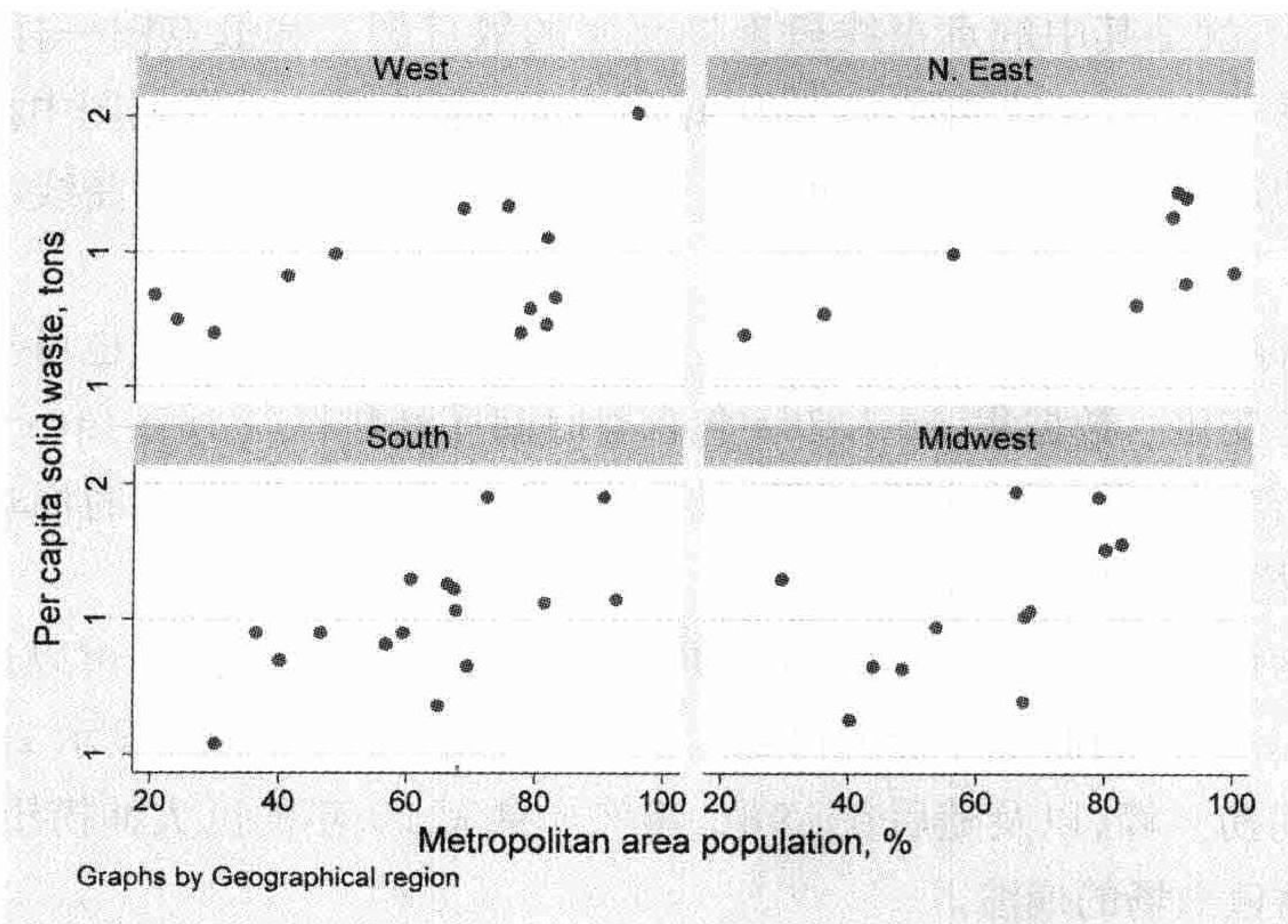


图 3.10

由 **graph matrix** 生成的散点图矩阵在多元分析中很有用。它们提供了各对变量交互关系的简洁展示,允许分析人员审视数据中可能影响统计建模的非线性、特异值或聚群(clustering)等问题的迹象。图 3.11 显示了 *states.dta* 数据中三个变量的散点图矩阵。

```
. graph matrix miles metro income waste, half msymbol(oh)
```

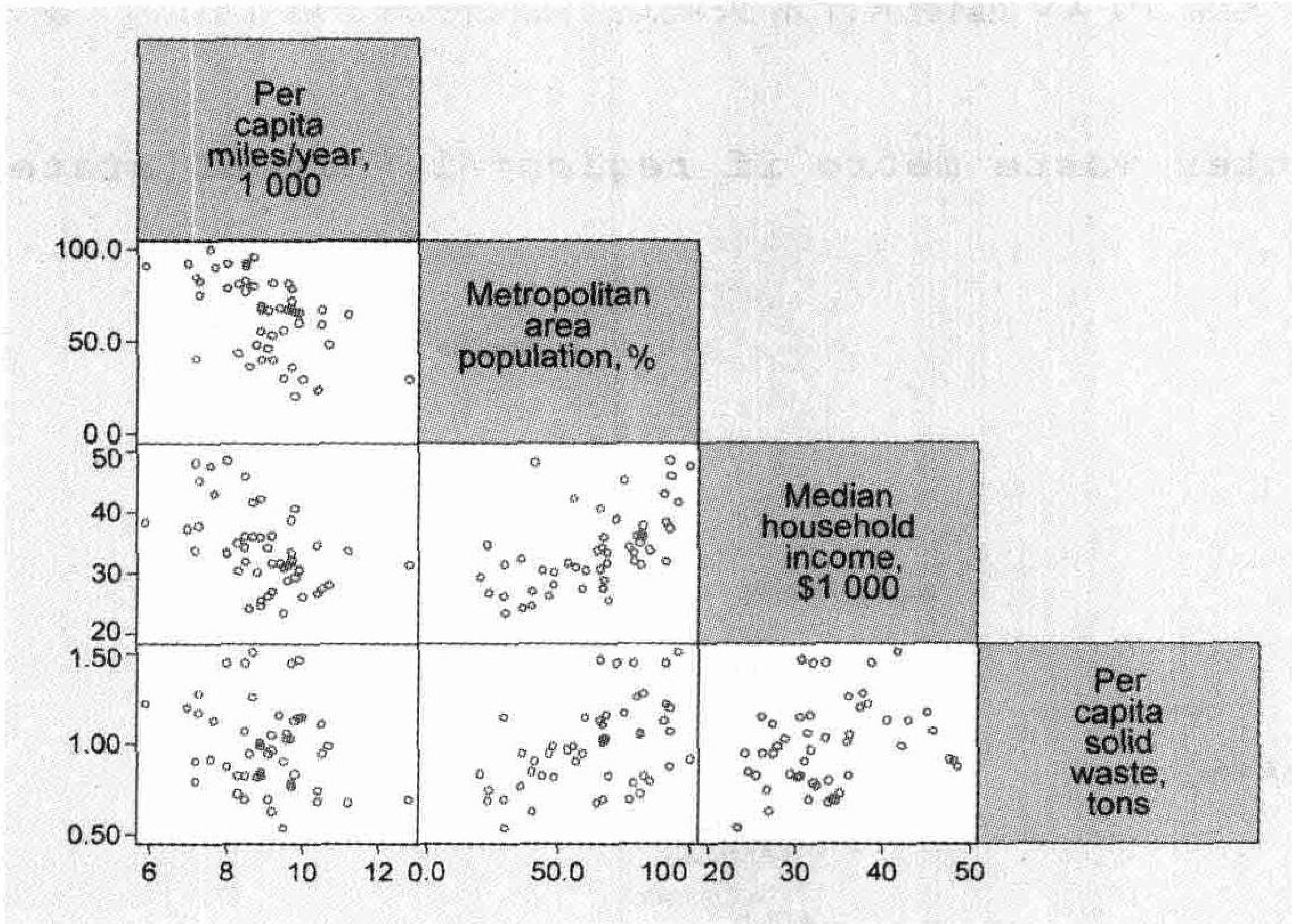


图 3.11

选项 **half** 指定图 3.11 应当只包括矩阵的下三角部分,因为上三角部分是与其对称的,所以对于许多研究目的而言都是多余的。选项 **msymbol(oh)** 要求对散点图采用我们想要的小空心圆圈作为标志。坐标轴的控制更为复杂,因为有多少变量就有多少坐标轴;请键入 **help graph_matrix** 查看具体细节。

当关注的变量中包含一个因变量或"结果"变量和数个自变量或"原因"变量时,最好将因变量列在 **graph matrix** 变量清单的最后。这样会形成整齐的一排横跨底部的因变量对自变量的散点图。

曲线标绘图

机械地看,曲线标绘图就是其中的点由线段连接起来的散点图。和散点图一样,曲线标绘图的不同类型也属于 Stata 功能强大的 **graph twoway** 族命令。散点图中控制添加坐标轴标签和标志的选项对曲线标绘图同样起作用。新的选项可以控制曲线本身的特征。

与散点图相比,曲线标绘图往往有不同的用法。比如,和时间图一样,它们也可描述一个变量随时间而发生的变化。数据集 *cod.dta* 包含着时间序列数据(time-series data),反映了纽芬兰北部鳕鱼渔场的不幸经历。该渔场曾经是世界最丰饶的渔场之一,但是主要由于过度捕捞,在 1992 年就崩溃了。

一个展示加拿大捕捞量(*canada*)和总的鳕鱼捕捞量(*cod*)的简单时间图可以通过画出这两个变量对年份(*year*)的曲线标绘图得到。图 3.12 就是这样的图,显示 1960 年代晚期国际性的过度捕捞高峰,以及随后在 1980 年代加拿大 10 年的巨大捕捞压力,最终导致 1992 年北部鳕鱼渔场的崩溃。

```
Contains data from C:\data\cod.dta
  obs:            38                          Newfoundland's Northern Cod
                                                fishery, 1960-1997
 vars:             5                          4 Jul 2005 15:02
 size:           684 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %8.0g                  Year
cod             float  %8.0g                  Total landings, 1000t
canada          int    %8.0g                  Canadian landings, 1000t
TAC             int    %8.0g                  Total Allowable Catch, 1000t
biomass         float  %9.0g                  Estimated biomass, 1000t
-------------------------------------------------------------------------------
Sorted by:  year
```

```
. graph twoway line cod canada year
```

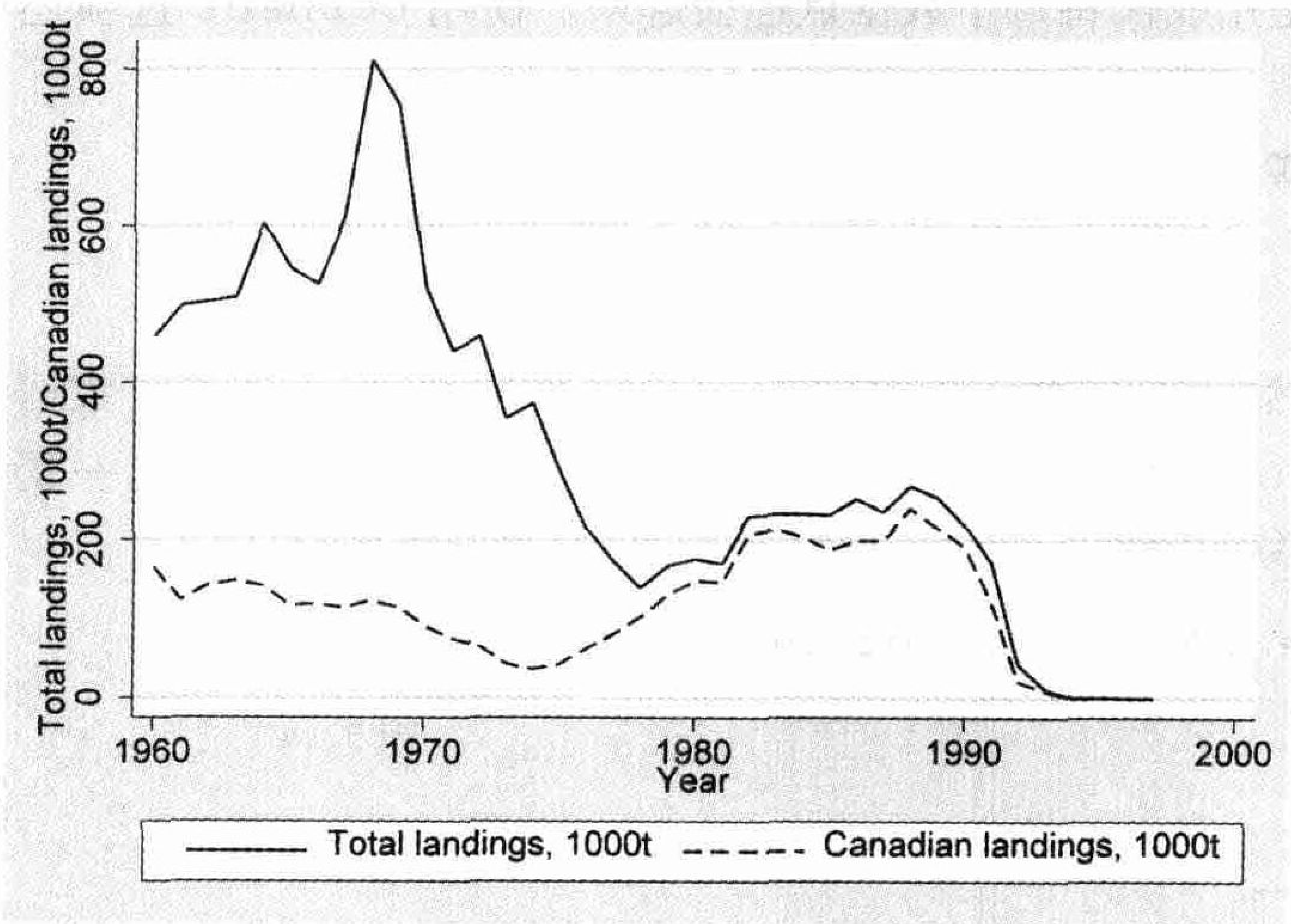


图 3.12

在图 3.12 中,Stata 自动地为第一个提到的 *y* 变量 *cod* 选择了一条实线,而为第二个变量 *canada* 选择了虚线。底部的图例说明了这些线条的含义。我们可以通过重新安排图例和去除多余的 *y* 轴标题来对该图加以改进,如图 3.13 所示。

```
. graph twoway line cod canada year, legend(label (1 "all nations")
     label(2 "Canada") position(2) ring(0) rows(2)) ytitle("")
```

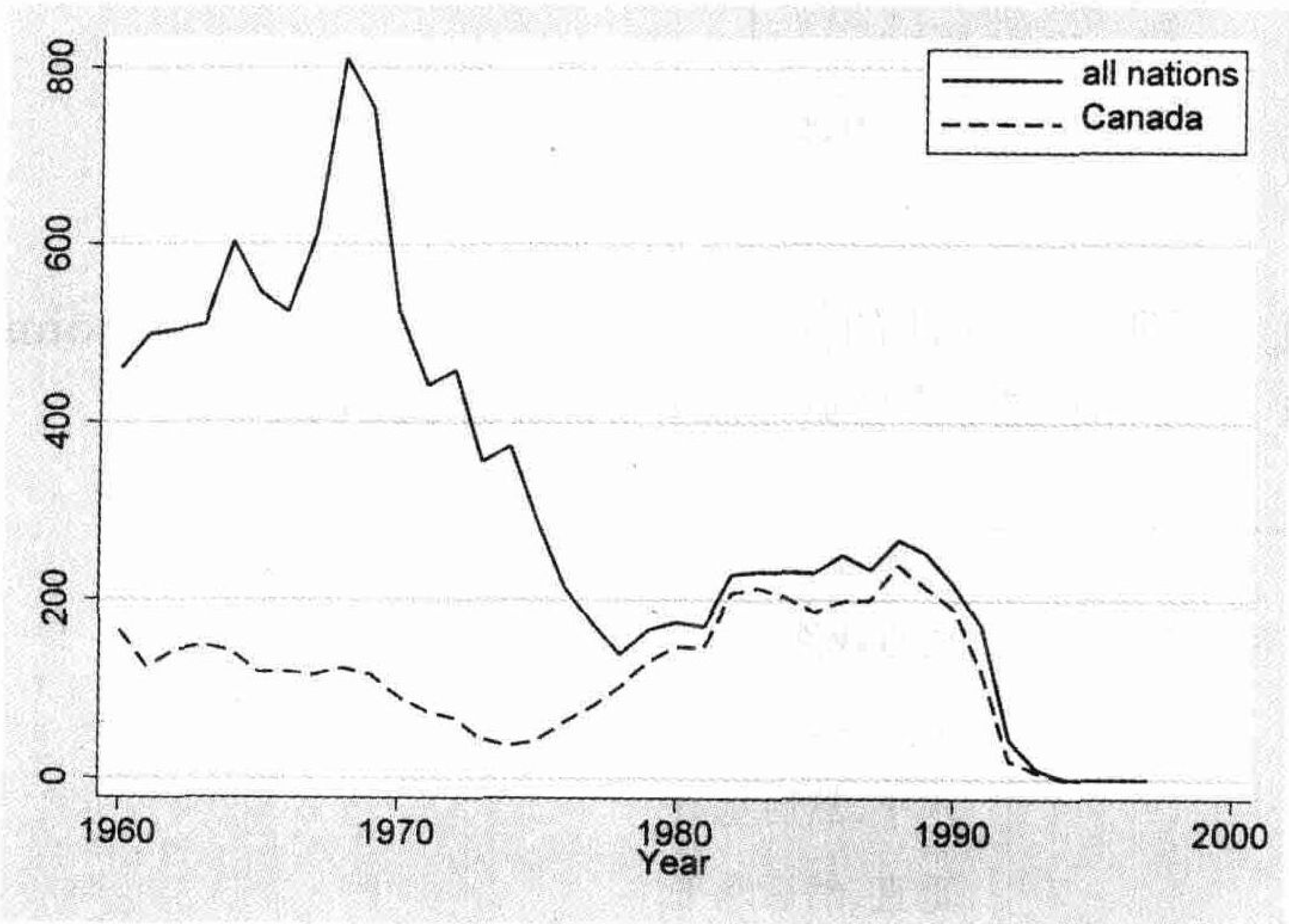


图 3.13

至于图3.13的 **legend**(图例)选项可以分解如下。请注意,所有这些子选项都出现在 **legend** 后面的括号之内。

label(1 "all nations")	为第一个提到的 y 变量添加标签"all nations"(所有国家)
label(2 "Canada")	为第二个提到的 y 变量添加标签"Canada"(加拿大)
position(2)	将图例放置在2点钟的位置上(右上角)
ring(0)	将图例放置在图中空白处
row(2)	将图例排列成两行

通过缩短图例标签并将其放置在图中的空白处,我们可以留下更多空间用于显示数据和创建更具吸引力和可读性的图形。**legend** 对其他带有图例的图形也起类似作用。键入 **help legend_option** 查看许多可用的子选项的清单。

图3.12 和图3.13 只是用线段将每个数据点连接起来。使用 **connect** 选项还可以使用好几种其他的连接类型。比如:

```
connect(stairstep)
```

或等价地用

```
connect(J)
```

将使各点以阶梯形(水平,然后垂直)线条连接起来。图3.14以取自 *cod.dta* 数据的政府所设总捕捞限量(government-set Total Allowable Catch, TAC)这一变量的阶梯形时间标绘图进行了举例说明。

```
. graph twoway line TAC year, connect(stairstep)
```

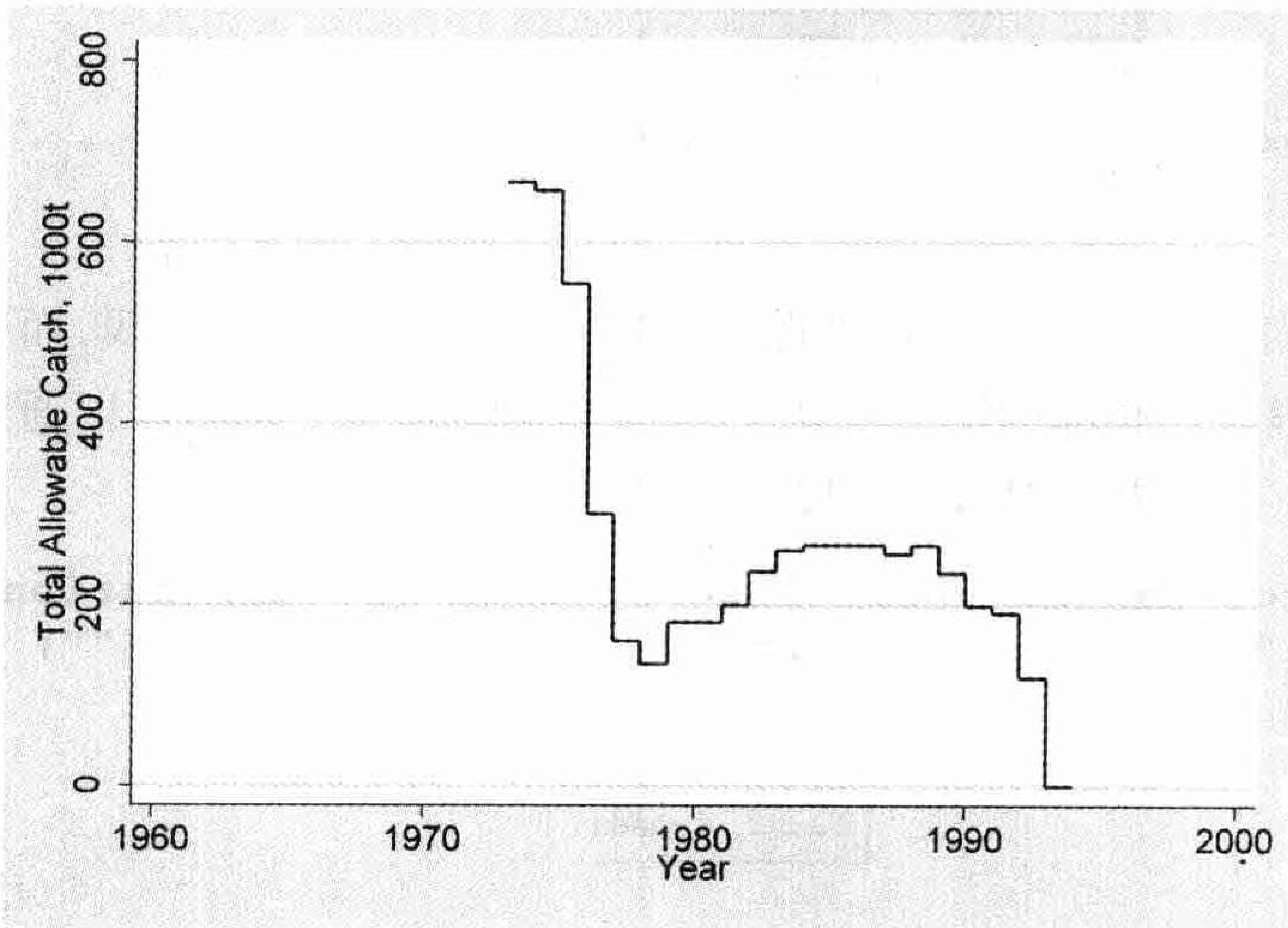


图3.14

其他的 **connect** 选项如下所列。默认的连线方式是直线,相当于选项 **connect(direct)** 或 **connect(l)**。更多内容,请参见 **help connectstyle**。

connect( )	缩 写	描 述
none	**i**	无连接
direct	**l**(字母"el")	用直线连接
ascending	**L**	笔直线,但只适合 $x[i+1] > x[i]$ 的情况
stairstep	**J**	水平,然后垂直
stepstair		垂直,然后水平

以下的图 3.15 重现了 *TAC* 的阶梯形标绘图,但是在坐标轴标签和标题上面作了一些改进。选项 **xtitle(" ")** 要求 *x* 轴不添加标题(因为“年份”是显而易见的)。我们在 *x* 轴上以两年为间隔添加刻度标志,*y* 轴则以 100 为间隔添加标签,而且 *y* 轴的标签按照横向而不是(默认的)纵向显示。

```
. graph twoway line TAC year, connect(stairstep)  xtitle("")
    xtick(1960(2)2000)  ytitle("Thousands of tons")
    ylabel(0(100)800, angle(horizontal)) xtitle("")
    clpattern(dash)
```

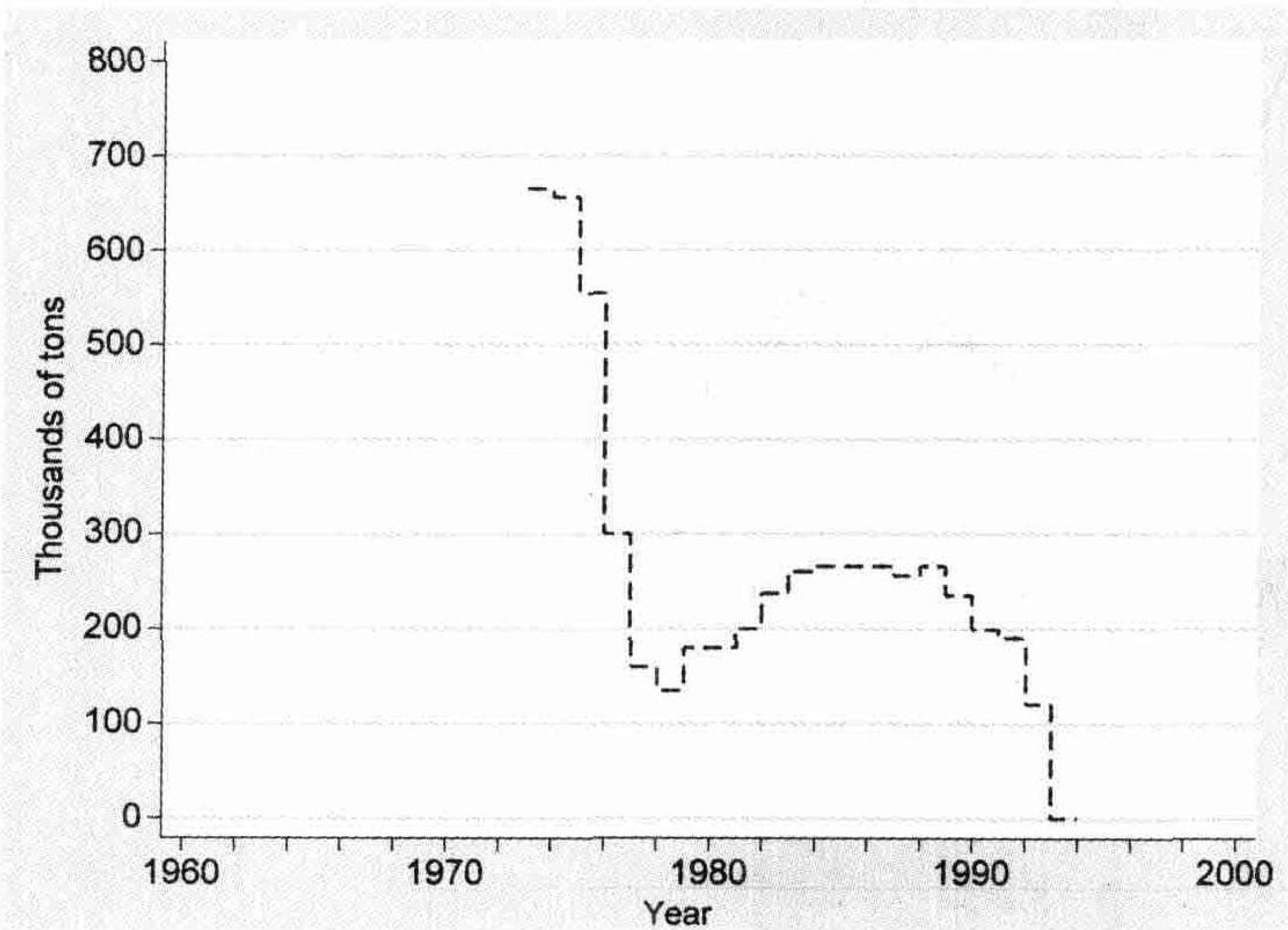


图 3.15

图 3.15 中,我们还使用了 **clpattern(dash)** 选项,要求使用虚线(dashed line)而不是任由 Stata 决定线条样式(pattern)(实线、虚线等)。线条样式的可能选项如下表所列(也可参见 **help linepatternstyle**)。

clpattern( )	描 述
solid	实线(solid line)
dash	虚线(dashed line)
dot	点线(dotted line)
dash_dot	点划线(dash then dot)
shortdash	短划线(short dash)
shortdash_dot	短划点线(short dash followed by dot)
longdash	长划线(long dash)
longdash_dot	长划点线(long dash followed by dot)
blank	不可见的线(invisible line)
formula	如,**clpattern(-.)**或 **clpattern(-..)**

在我们转到其他的例子和类型之前,以图 3.16 将本节讨论到的三个变量结合起来创建为一个图形,展示了北部鳕鱼渔场的悲剧。请注意 **connect( )**、**clpattern( )** 和 **legend( )** 选项在涉及三个变量的情况下是如何起作用的。

```
. graph twoway line cod canada TAC year, connect(line line stairstep)
    clpattern(solid longdash dash) xtitle("") xtick(1960(2)2000)
    ytitle("Thousands of tons") ylabel(0(100)800, angle(horizontal))
    xtitle("") legend(label (1 "all nations") label(2 "Canada")
    label(3 "TAC") position(2) ring(0) rows(3))
```

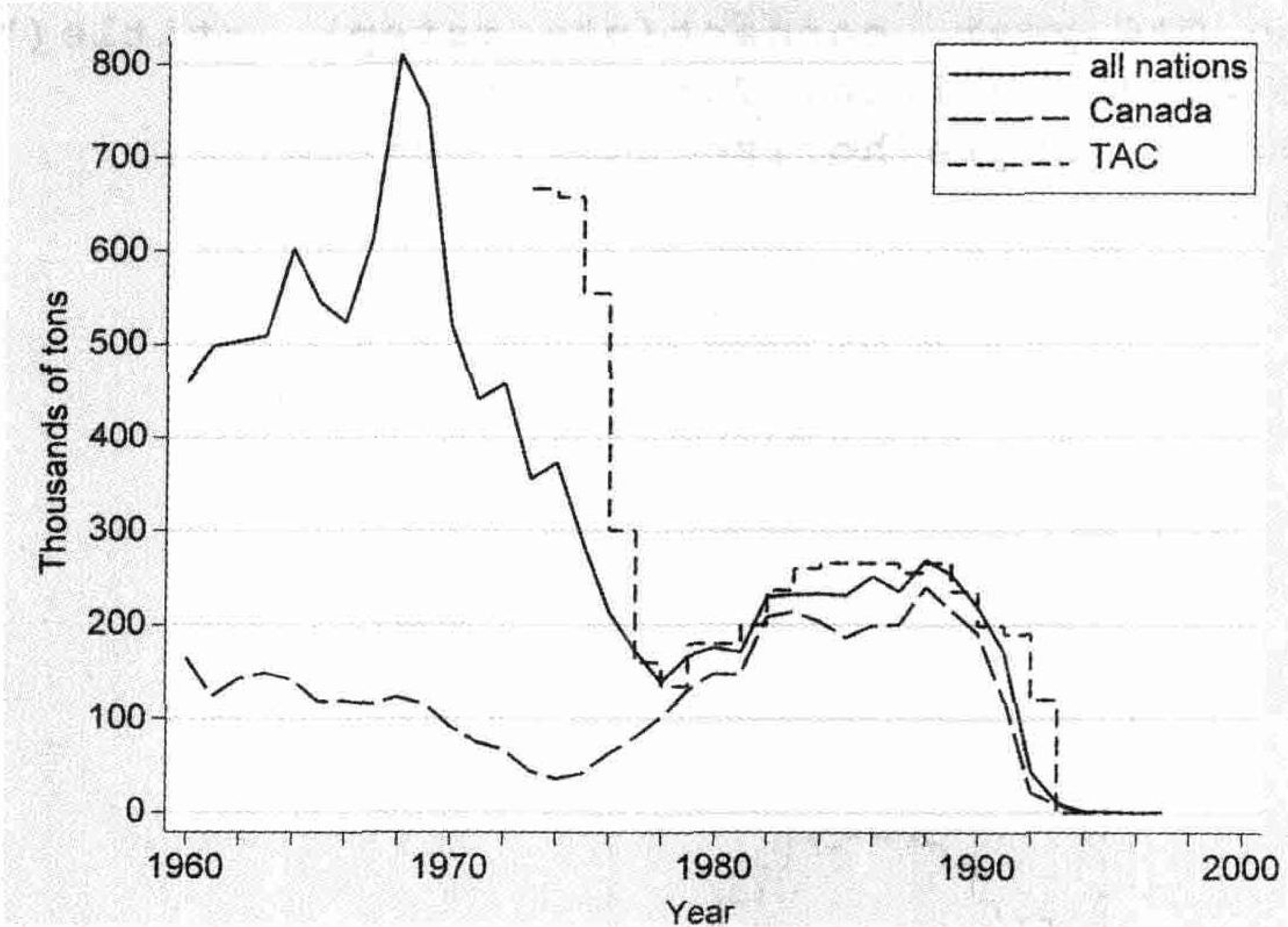


图 3.16

连线标绘图

在上节的曲线图中,数据点是不可见的,我们只看到连线。命令 **graph twoway connected** 创建连线标绘图(connected-line plot),图中的数据点由散点图记号加以标志。前面对 **graph twoway scatter** 描述过的标志记号选项以及对 **graph twoway line** 描述过的连线样式选项也都可以应用于 **graph twoway connected**。图 3.17 展示了一个默认状态下的例子,取自 *cod.dta* 数据中测量鳕鱼单位面积数量的变量(*biomass*)的连线时间标绘图。

```
. graph twoway connected biomass year
```

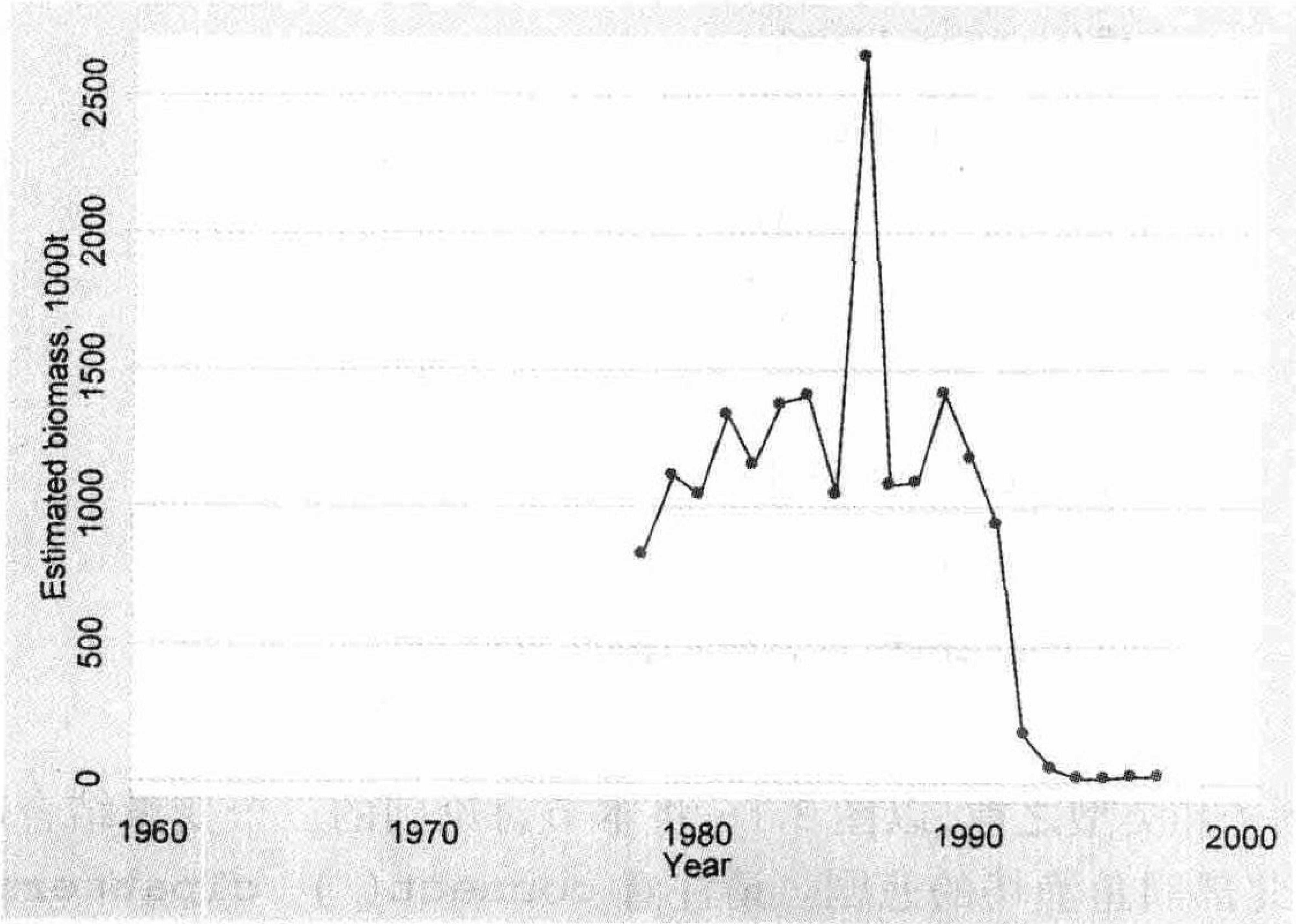


图 3.17

数据中只包含1978—1997年的单位面积数量值，这导致图3.17出现大片空白。`if`选择条件为我们限定年份取值范围提供了可能。下面的图3.18实现了这点，它还对图形进行了装饰，以展示对标志记号、线条样式、坐标轴和图例上的控制。由于鳕鱼捕捞量和单位面积数量同时位于同一图形中，因此我们看到鳕鱼单位面积数量在1980年代晚期开始骤降，即这场危机的发生实际上比官方承认的要早好几年。

```
. graph twoway connected bio cod year if year > 1977 & year < 1999,
      msymbol(T Oh) clpattern(dash solid) xlabel(1978(2)1996)
      xtick(1979(2)1997) ytitle("Thousands of tons") xtitle("")
      ylabel(0(500)2500, angle(horizontal))
      legend(label(1 "Estimated biomass") label(2 "Total landings")
      position(2) rows(2) ring(0))
```

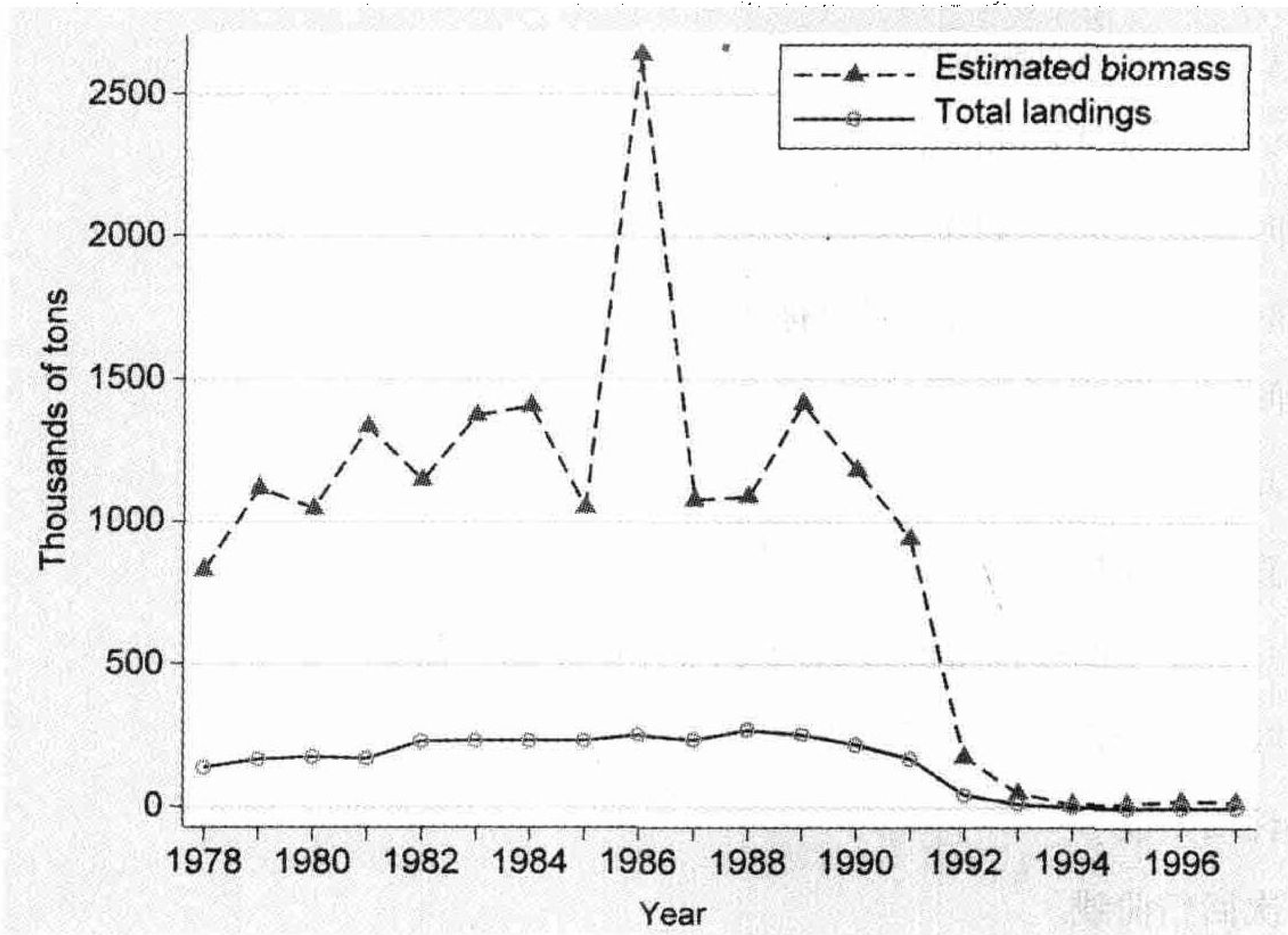


图3.18

其他类型的二维标绘图

除了基本的曲线标绘图和散点图之外，**graph twoway**命令还包含许多其他的类型。下表列出了可能的情形：

graph twoway	描 述
scatter	散点图
line	曲线标绘图
connected	连线标绘图
scatteri	即时参数(immediate arguments)的散点图(数据在命令行中给定)
area	线图加以区域着色
bar	二维条形图(不同于**graph bar**)
spike	二维芒线图
dropline	垂线图(点向特定数值做垂直或水平的连线)
dot	二维点标绘图(不同于**graph dot**)
rarea	全距图，将高端和低端值之间区域着色
rbar	在高端和低端值之间添加条形的全距图
rspike	在高端和低端值之间添加芒线的全距图
rcap	带有两端被戴帽的芒线的全距图

续表

graph twoway	描 述
rcapsym	带有两端被加有记号的芒线的全距图
rscatter	带有散点标志的全距图
rline	带线条的全距图
rconnected	带线条和标志的全距图
pcspike	带芒线的成对坐标图(paired-coordinate plot)
pccapsym	带有两端被加有记号的芒线的成对坐标图
pcarrow	带箭头的成对坐标图
pcbarrow	带双箭头的成对坐标图
pcscatter	带记号的成对坐标图
pci	即时参数的带芒线的成对坐标图
pcarrowi	即时参数的带箭头的成对坐标图
tsline	时间序列标绘图
tsrline	时间序列全距图
mband	以直线段连接波段内(x,y)的交叉中位数
mspline	以立方样条曲线连接波段内(x,y) 的交叉中位数
lowess	LOWESS(局部加权的散点图修匀)曲线
lfit	线性回归线
qfit	二次回归曲线
fpfit	分式多项式标绘图
lfitci	带置信区间的线性回归线
qfitci	带置信区间的二次回归曲线
fpfitci	带置信区间的分式多项式标绘图
function	函数的曲线标绘图
histogram	直方图
kdensity	内核密度标绘图

控制线条样式、标志记号等等的常用选项适用于所有的 **twoway** 命令。有关具体命令的更多信息,请键入 **help twoway_mband** 或 **help twoway_function** 等(使用上述名称中的任何一个)咨询。请注意,**graph twoway bar** 是不同于 **graph bar** 命令的。类似地,**graph twoway dot** 也不同于 **graph dot**。类似于散点图或曲线图,**twoway** 版的命令提供了一个测量型的 y 变量对另一个测量型的 x 变量制图的不同方法。但是,非 twoway 版的命令则提供了对一个或更多个测量型的 y 变量对一个或更多个 x 变量的不同类别的概要统计量(如平均数或中位数)制图的方法。**twoway** 版的命令因此是比较专门的,尽管(和所有的 **twoway** 图一样)它们可以和其他的 **twoway** 图形叠并在一起,以取得更复杂的图形效果。

正如本章稍后描述的那样,这些图形类别中的许多种都能通过叠并两幅或更多幅简

单图形来构成复合图形，这是极为有用的。其他一些则产生精致的独立图形。比如，图 3.19 显示了一幅纽芬兰渔场鳕鱼捕获量的区域图。

```
. graph twoway area cod canada year, ytitle("")
```

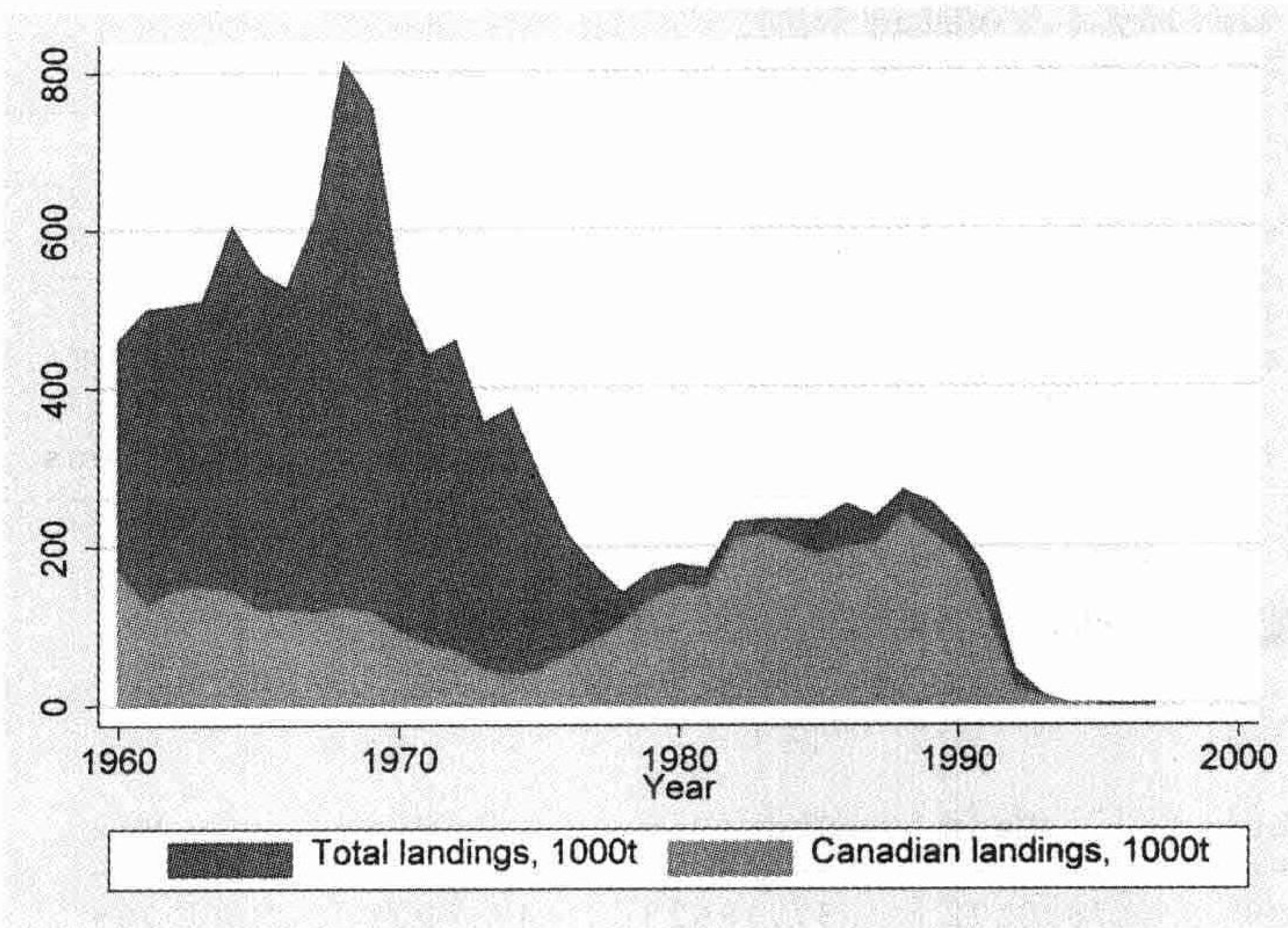


图 3.19

区域图中着色(shading)和其他图形中的区域着色都可以通过选项 **bcolor** 进行控制。请键入 **help colorstyle** 查看可选的颜色，其中也包括灰度(gray scale)。最深的灰度(gs0)实际上就是黑色。最浅的灰度(gs16)是白色。其他灰度值位于这两者之间。比如，图 3.20 显示了上图的浅灰度版。

```
. graph twoway area cod canada year, ytitle("") bcolor(gs12 gs14)
```

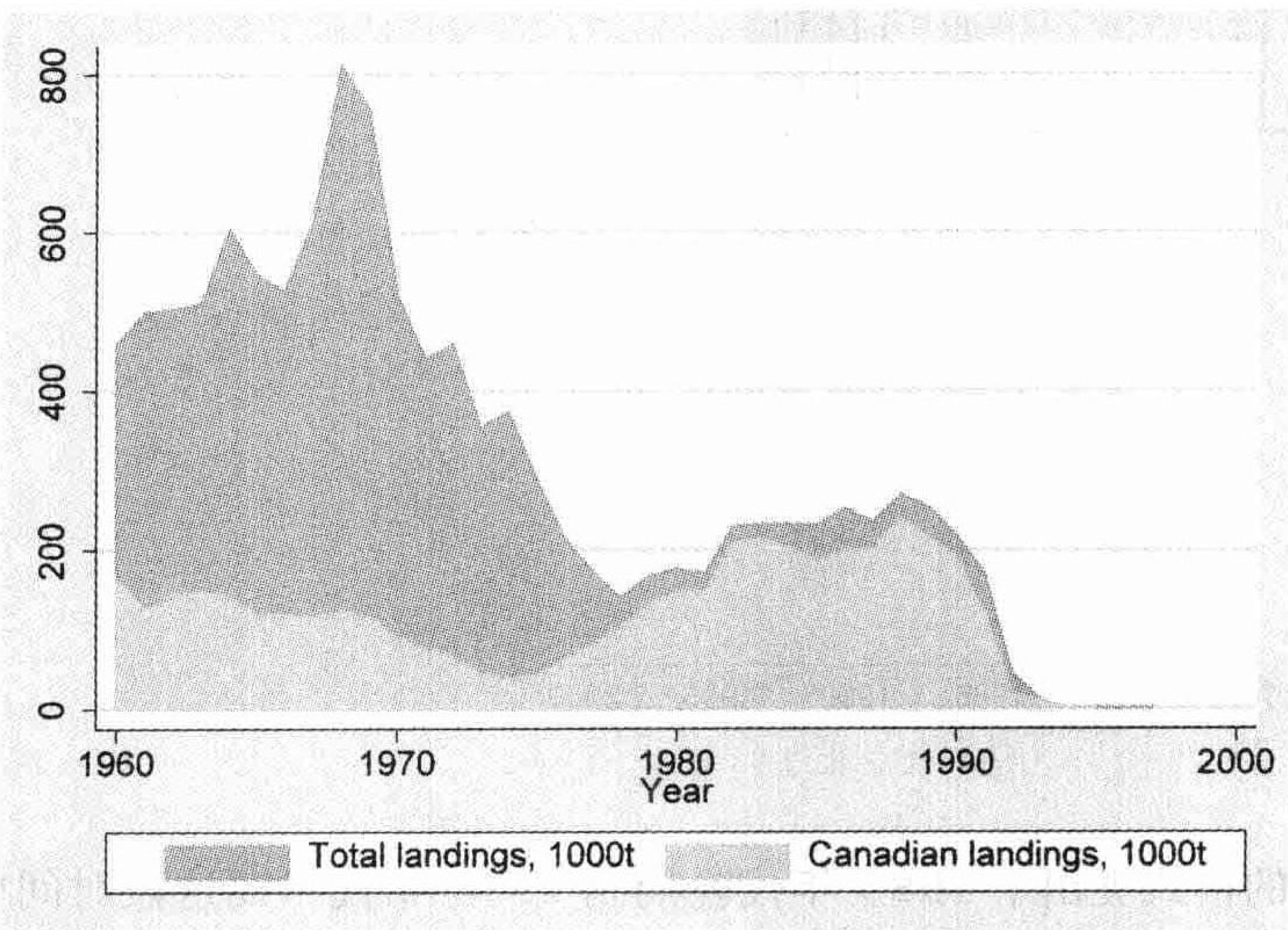


图 3.20

异常的冷空气或海洋状况在纽芬兰渔场的灾难中起着次要的作用，这一灾难不仅波及北部鳕鱼渔场，还影响到其他物种和总量。比如，邻近的圣劳伦斯湾的主要鱼种在这一时期也出现了下降(Hamilton, Haedrich and Duncan, 2003)。数据集 *gulf.dta* 描述了环境和北部湾鳕鱼捕捞量(原始数据取自 DFO, 2003)。

```
Contains data from C:\data\gulf.dta
  obs:            56                          Gulf of St. Lawrence
                                                environment and cod fishery
 vars:             7                          10 Jul 2005 11:51
 size:         1344 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
winter          int    %8.0g                  Winter
minarea         float  %9.0g                  Minimum ice area, 1000 km^2
maxarea         float  %9.0g                  Maximum ice area, 1000 km^2
mindays         byte   %8.0g                  Minimum ice days
maxdays         byte   %8.0g                  Maximum ice days
cil             float  %9.0g                  Cold Intermediate Layer
                                                temperature minimum, C
cod             float  %9.0g                  N. Gulf cod catch, 1000 tons
-------------------------------------------------------------------------------
Sorted by:  winter
```

这些年间,年份冰面覆盖面积最大值平均为173 017平方公里。

```
. summarize maxarea

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     maxarea |        38    173.0172    37.18623    47.8901   220.1905
```

图3.21使用这一平均数(173千)作为芒线图的基线(base),图中基线上下的芒线分别表示高于或低于平均冰面覆盖量。选项 **yline(173)**要求在173的位置画一条水平线。

```
. graph twoway spike maxarea winter if winter > 1963, base(173)
    yline(173) ylabel(40(20)220, angle(horizontal))
    xlabel(1965(5)2000)
```

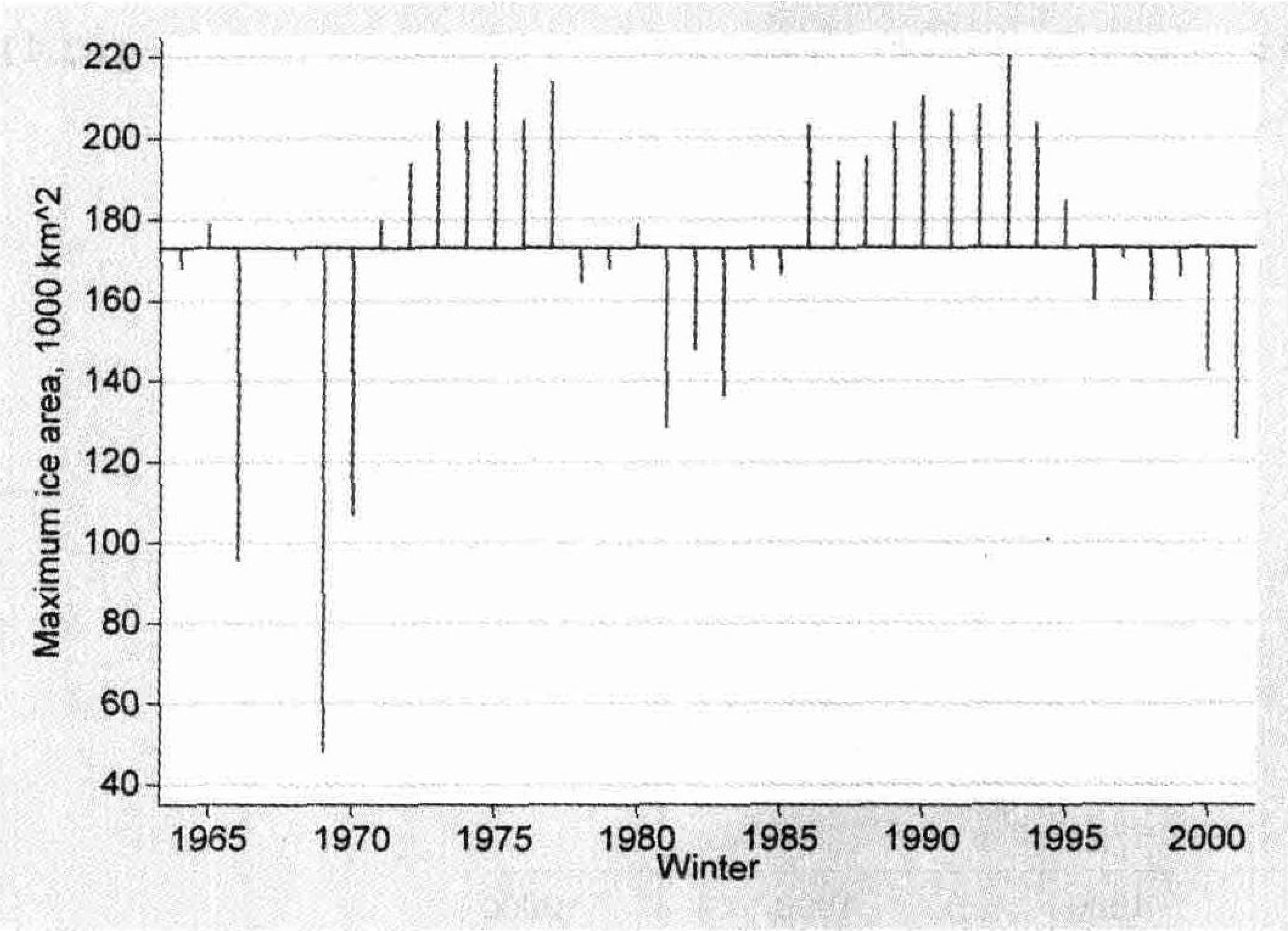


图3.21

图3.21中 **base( )**的格式突出了1980年代晚期到1990年代早期这段时间中连续发生的异常严冬(最大冰面覆盖面积高于平均水平),而这段时间大约正是纽芬兰发生渔业危机的时候。我们也能看到1980年代早期及以前有一段温和的冬天,以及近年来回暖趋势的迹象。

图3.22中,对同一数据的不同视角采用lowess回归对时间序列进行修匀。波段宽度选项 **bwidth(.4)**设定基于修匀数据点的曲线,这些数据点根据包含40%样本点的移动波段内的加权回归计算得到。如果缩小波段宽度,比如,采用 **bwidth(.2)**,即用20%的数据的波段宽度来修匀,将给我们一条更为参差不齐的曲线,它修匀得更少因而更接近于原

始数据。默认状态采用 **bwidth(.8)** 这样更高的波段宽度,因此修匀的幅度更大。不管所选的波段宽度是多少,指向 x 取值任一极端值的修匀点都必须根据越来越窄的波段进行计算,因此将表现出更少的修匀。第 8 章包含更多有关 lowess 修匀的内容。

```
. graph twoway lowess maxarea winter if winter > 1963, bwidth(.4)
     yline(173) ylabel(40(20)220, angle(horizontal))
     xlabel(1965(5)2000)
```

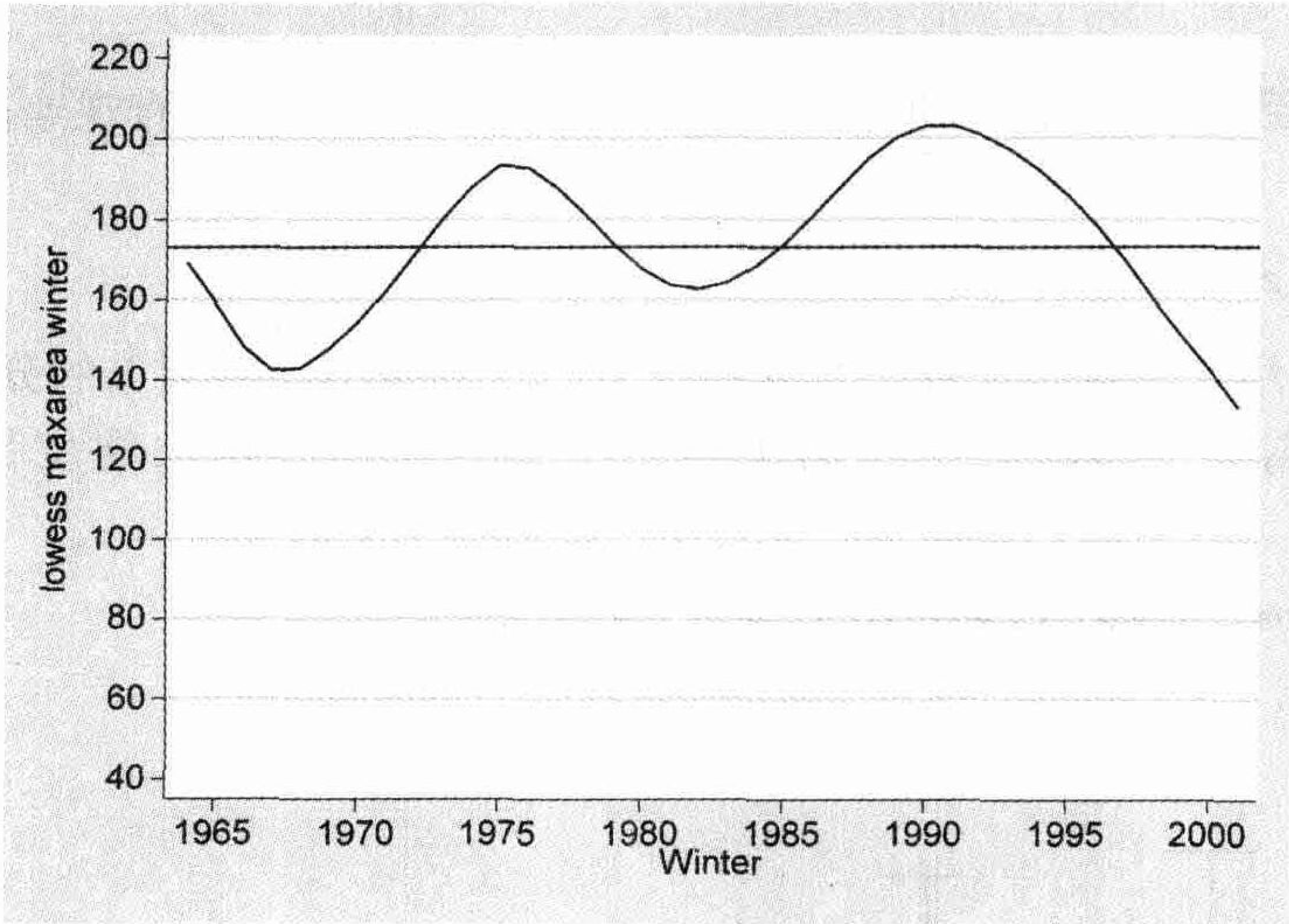


图 3.22

全距图使用条形(bars)、芒线(spikes)或着色区域(shaded areas)连接 x 每一水平上 y 值的高端与低端。每日股市价格常常采用这种方式得以画出。图 3.23 显示了数据 *gulf.dta* 中冰面覆盖面积的最小值(*minarea*)和最大值(*maxarea*)这两个变量画出的两端戴帽的芒线全距图。

```
. graph twoway rcap minarea maxarea winter if winter > 1963,
     ylabel(0(20)220, angle(horizontal)) ytitle("Ice area, 1000 km^2")
     xlabel(1965(5)2000)
```

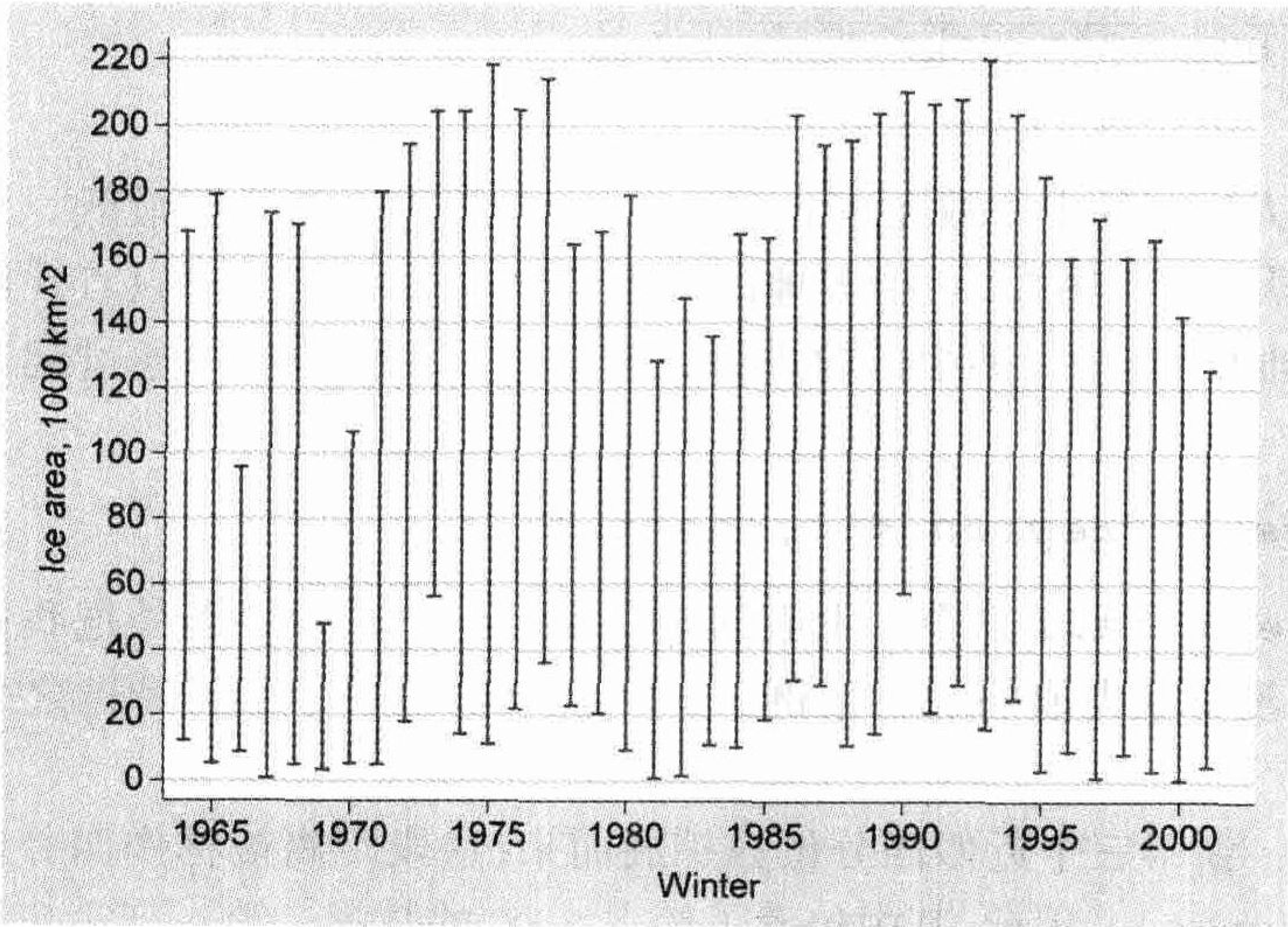


图 3.23

这些例子绝没有穷尽二维图的可能性。其他的应用贯穿于本书全书之中。在本章的稍后部分,我们将看到把两个或更多的二维图叠并成一个单一图形的例子。

箱线图

箱线图(box plot)直观地提供了有关中心、散布、对称和特异值的信息。想得到一幅单一的箱线图,请键入以下格式的命令:

```
. graph box y
```

如果几个不同变量具有大致相似的测量尺度,我们可以通过以下格式的命令直观地比较它们的分布:

```
. graph box w x y z
```

箱线图最为常见的应用之一就是比较一个变量在另一个变量不同类别上的分布。图3.24 比较了来自数据集 *states.dta* 的美国四个地区之间各州大学生比例(*college*)的分布。

```
. graph box college, over(region) yline(19.1)
```

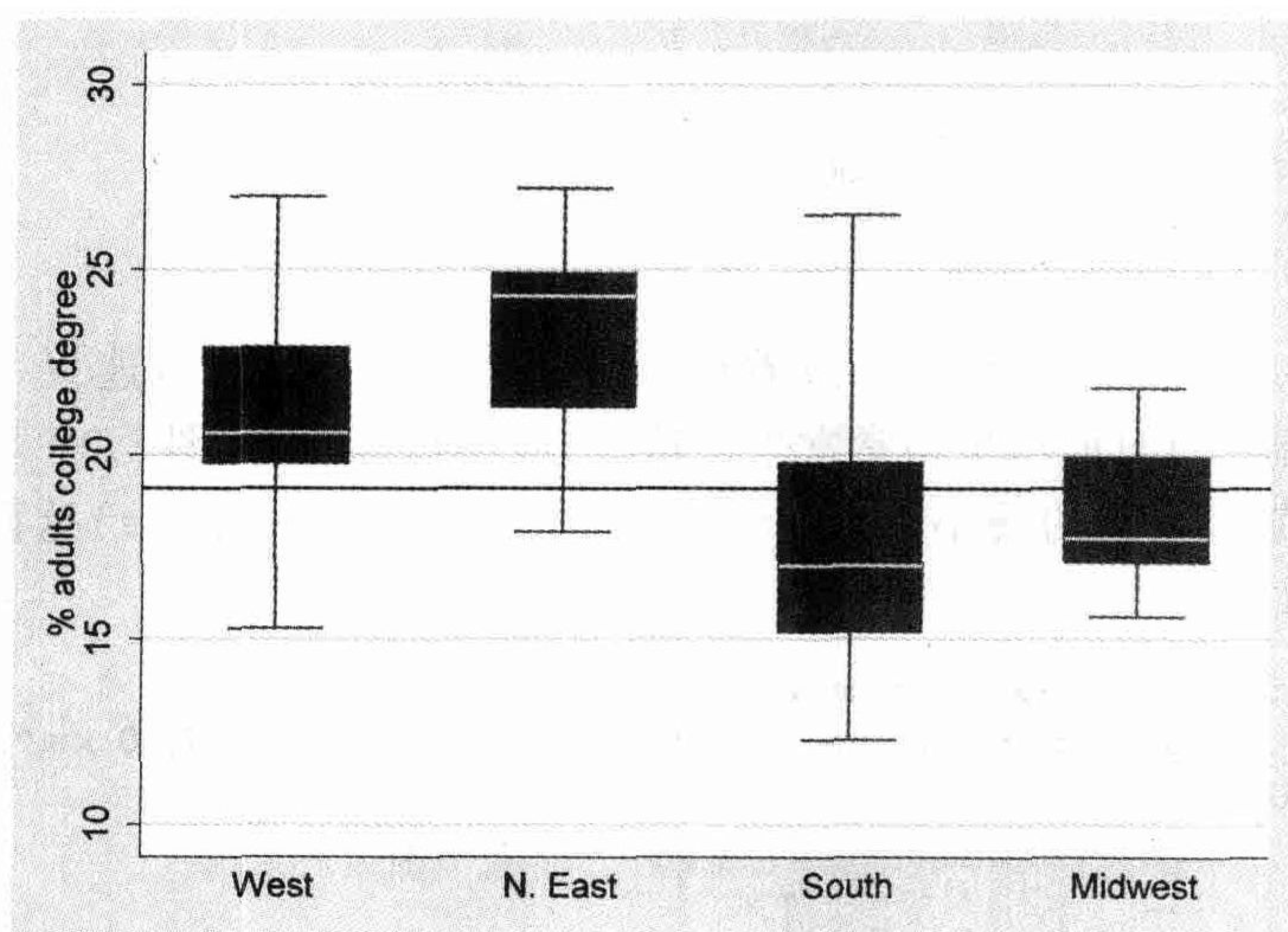


图 3.24

具有大学学历的成年人比例的中位数在东北部地区最高、而在南部地区最低。另一方面,南部各州之间的差异更大。图 3.24 中各地区的中位数(盒内的线)可以直观地与由选项 **yline(19.1)** 添加的 50 个州的中位数进行比较。这一中位数值可通过键入以下命令得到:

```
. summarize college if region < ., detail
```

第 4 章描述了 **summarize, detail** 命令。上面的 **if region <.** 选择条件把我们的分析限定在那些 *region* 没有缺失值的观测案例上;也就是说,除了华盛顿特区之外的所有各州。

箱线图中的箱子从第一到第三个近似四分位数扩展而来,这段距离被称作四分位距(IQR,interquartile range)。因此,其中包含了约占 50% 的数据。第一到第三个四分位距之外大于 1.5 倍 IQR 的那些观测案例被定义为特异值,它们在箱线图中被单独地一一画出。图 3.24 中的四个分布都没有出现特异值。Stata 的箱线图用与 **summarize, detail** 相同的方式来定义四分位数。这与(第 4 章)用于计算字母取值显示的 **lv** 的"第四"值并不是同样的近似方法。有关四分位数近似值及其在辨别特异值中的作用的更详细

内容,请参见:Frigge, Hoaglin, and Iglewicz(1989),以及 Hamilton(1992b)。

许多选项控制着箱线图中箱子的外观、着色和细节,有关的清单,请见 **help graph_box**。图 3.25 使用来自数据 *states.dta* 中的人均能源消费量(*energy*)示范了这些选项中的一部分,还采用了 **graph hbox** 的水平(horizontal)布局。选项 **over(region, sort(1))** 要求根据箱子对第一个提到的(本例是唯一的)*y* 变量上的中位数作升序排列。**intensity(30)** 用以控制箱子中着色的亮度,这里的设定值比图 3.24 中默认状况要略微低一些(即更浅)。图 3.25 中有悖视觉习惯地以垂直线标志了总的中位数(320),因此要求采用了选项 **yline**、而不是 **xline**。

```
. graph hbox energy, over(region, sort(1)) yline(320) intensity(30)
```

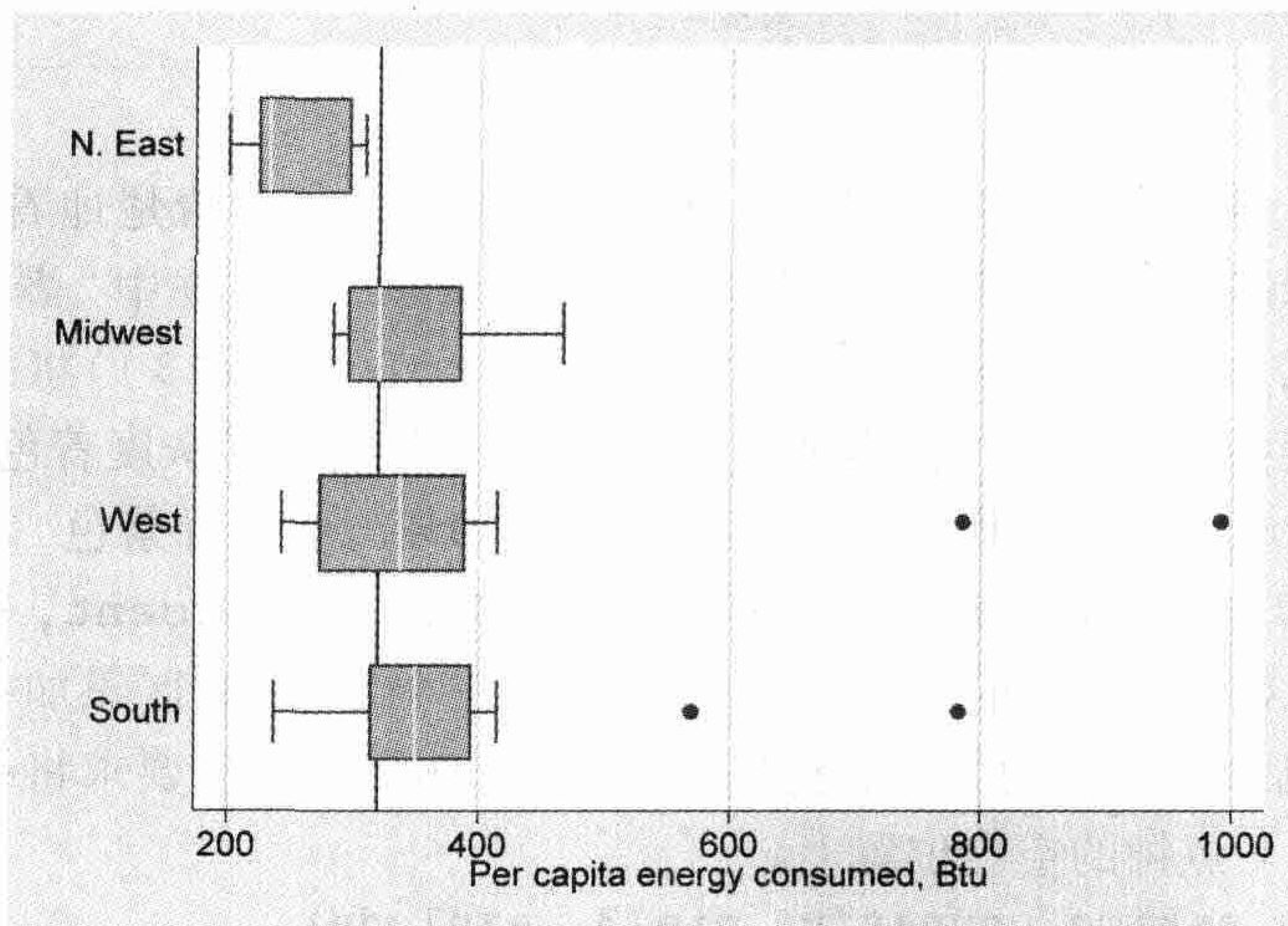


图 3.25

图 3.25 中的能源箱线图不但说明了中位数之间的差异,而且还说明存在着特异值,比如,西部和南部地区有四个极高消费量的州。略微加以深究,我们就发现这些都是产油的州:怀俄明、阿拉斯加、德克萨斯和路易斯安娜。箱线图擅长于引起对特异值的注意,而特异值在其他统计分析中则很容易被忽略(并且常常引起麻烦)。

饼 图

尽管对分析工作价值很小,但饼图(pie chart)仍是"图形表达"的流行工具。Stata 的基本饼图命令具有如下格式:

```
. graph pie w x y z, pie
```

这里[2],变量 *w*、*x*、*y* 和 *z* 都是以相似的单位测量某事物的量(比如,全都以美元、小时或人为单位)。

有关阿拉斯加人口民族构成的数据集 *AKethnic.dta* 提供了一个例子。阿拉斯加的本地土著民族人口分成三个宽泛的文化或语言群体:阿留申族(Aleut)、印地安族(Indian)和爱斯基摩族(Eskimo)。取自 1990 年美国人口普查的民族变量 *aleut*、*indian*、*eskimo* 和 *nonnativ*(非土著)分别是每个群体的人口数。该数据只包含三条代表三种规模的社区观测案例:即 10 000 及以上人口的市、1 000 到 10 000 人口的

[2]【译注:此句命令的选项 **pie** 后必须加括号及有关定义,如 **pie(#,explode)**。】

城镇,以及少于1 000 人的乡村。

```
Contains data from C:\data\AKethnic.dta
  obs:             3                          Alaska ethnicity 1990
 vars:             7                          4 Jul 2005 12:06
 size:            63 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
comtype         byte   %8.0g       popcat     Community type (size)
pop             float  %9.0g                  Population
n               int    %8.0g                  number of communities
aleut           int    %8.0g                  Aleut
indian          int    %8.0g                  Indian
eskimo          int    %8.0g                  Eskimo
nonnativ        float  %9.0g                  Non-Native
-------------------------------------------------------------------------------
Sorted by:
```

该州人口的大部分为非土著人,这一点可以从一个饼图(图3.26)中清楚地看到。为了强调,选项 **pie(3, explode)** 使第三个提到的变量 *eskimo* 从饼中"爆出"(exploded)。为了与更小的土著人口相比较,用选项 **pie(4, color(gs13))** 将第四个提到的变量 *nonnativ* 着色成浅灰色。(在本书中,我们的例子只使用了灰度着色,但是请记住还有其他诸如 **color(blue)**(蓝色)或 **color(cranberry)**(青紫色)等可能的选择。请键入 **help colorstyle** 查询有关清单。) **plabel(3 percent, gap(20))** 要求在距中心20个相对半径单位的位置对 *eskimo*(第三个变量)扇区添加一个百分比标签。我们看到阿拉斯加人口中约8%是爱斯基摩人。**legend** 选项要求将一个四行的方框图例放置在图中11点钟的空白位置上。

```
. graph pie aleut indian eskimo nonnativ, pie(3, explode)
      pie(4, color(gs13)) plabel(3 percent, gap(20))
      legend(position(11) rows(4) ring(0))
```

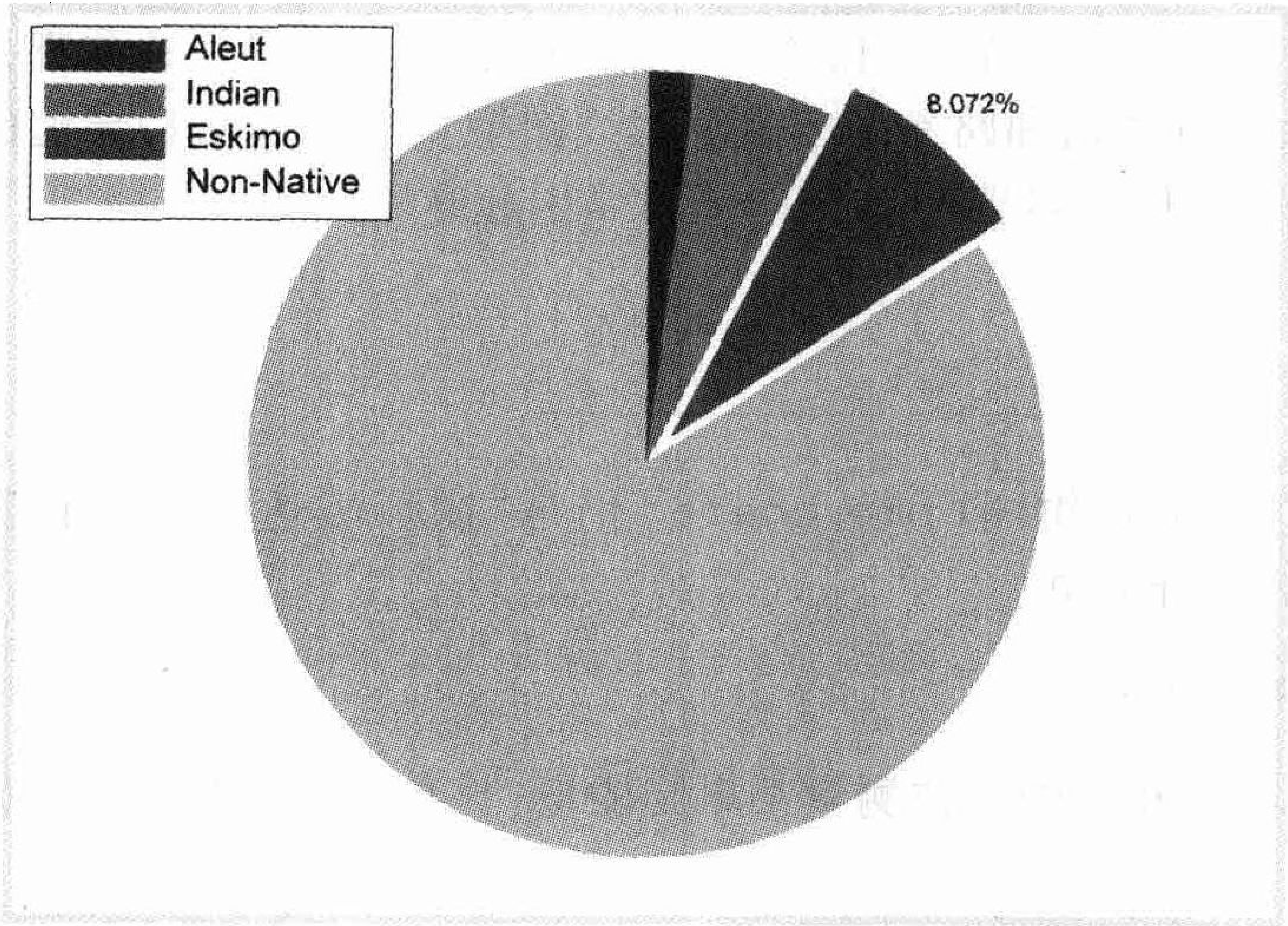


图3.26

非土著人是图3.26中的人口主体,但是如果我们通过增加选项 **by(comtype)** 对不同社区类型分别画出饼图的话,就会出现新的细节(图3.27)。选项 **angle0( )** 设定饼中第一个扇区(slice)的起始角度(angle)。图3.27中第一扇区的起始角度被设定为0度(水平方向),标签在这一角度时更易读。图形显示出,尽管土著人只是阿拉斯加城市人口中的小部分,但是他们构成了乡村人口的大多数。尤其是爱斯基摩人占了村民中的较大部分,即所有村民人口的35%,而且在一些地方甚至超过了90%。这使得阿拉斯

加乡村具有不同于阿拉斯加城市的特征。

```
. graph pie aleut indian eskimo nonnativ, pie(3, explode)
     pie(4, color(gs13)) plabel(3 percent, gap(8))
     legend(rows(1)) by(comtype) angle0(0)
```

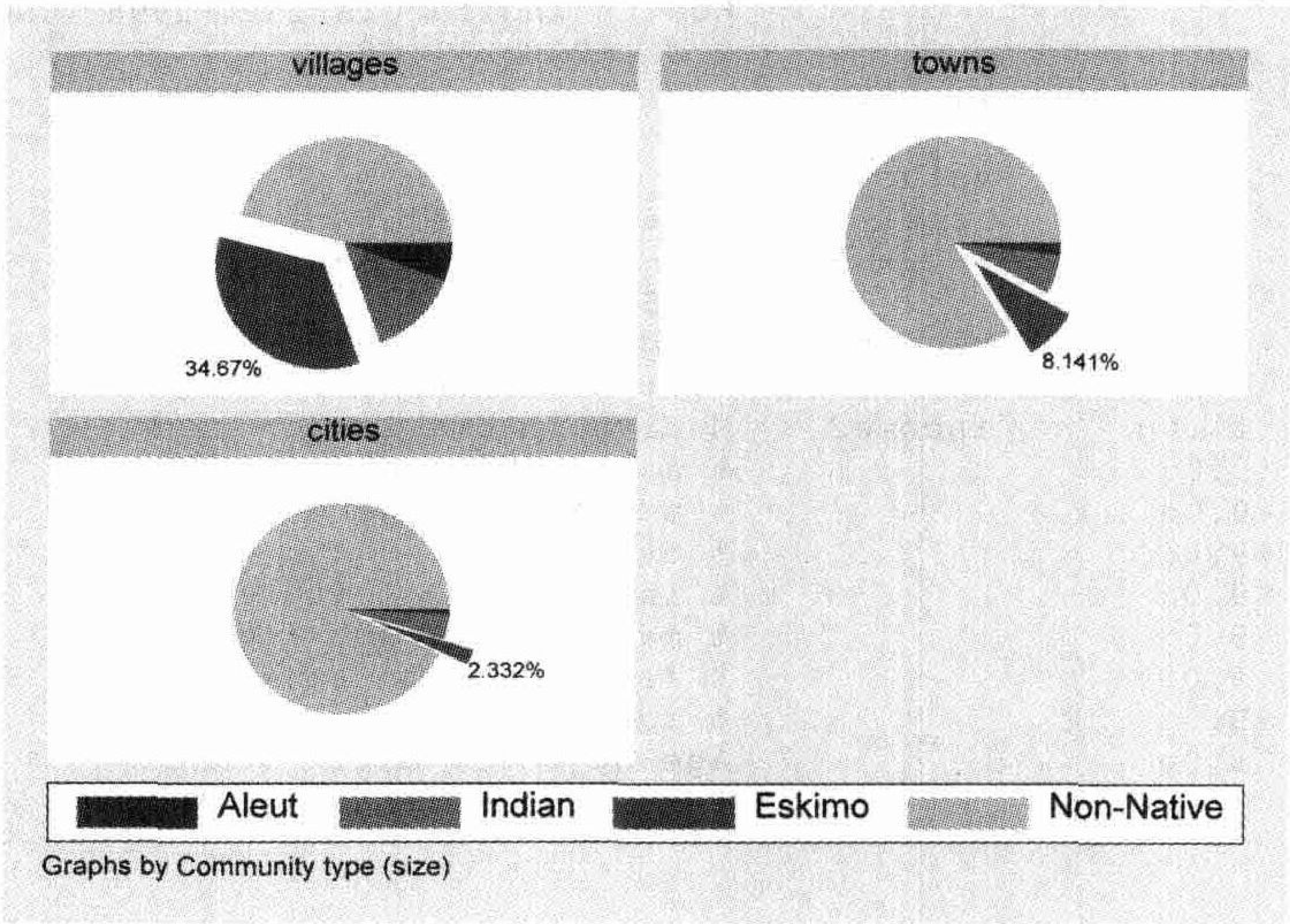


图 3.27

条形图

尽管条形图(bar chart)比箱线图包含更少的信息,但是它们仍然为比较平均数、中位数、合计数或计数等多种概要统计提供了简单而又多样化的展示。比如,要想得到显示 *y* 在 *x* 不同类别上的平均数的直方条,键入:

```
. graph bar (mean) y, over(x)
```

为了显示 *x2* 的不同类别内又在 *x1* 不同类别上的 *y* 的合计(sum)的水平方条,键入:

```
. graph hbar (sum) y, over(x1) over(x2)
```

条形图可以显示以下任何一个统计量:

mean	平均数(默认设定;如不指定统计量类型的话,就会输出平均数)
sd	标准差
sum	合计数
rawsum	忽略随意设定权数的合计数
count	具有非缺失值的观测案例数
max	最大值
min	最小值
median	中位数
p1	第一百分位数
p2	第二百分位数
iqr	四分位距

这一可用的概要统计量清单和 **collapse** 命令(见第 2 章)对应的清单相同,也和 **graph dot** (下一节)与 **table** (第 4 章)等许多其他命令对应的清单相同。

数据集 *statehealth.dta* 包含了美国各州的更多数据信息，它将来自1990年普查的社会经济指标和来自疾病控制中心(CDC，2003)对1994—1998年期间若干平均健康风险指标合并在一起。

```
Contains data from C:\data\statehealth.dta
  obs:            51                          Health indicators 1994-1998 (CDC)
 vars:            12                          9 Jul 2005 11:56
 size:          3315 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
state           str20  %20s                   US State
region          byte   %9.0g       region     Geographical region
income          long   %10.0g                 Median household income, 1990
income2         float  %11.0g      income2    Median income low or high
high            float  %9.0g                  % adults HS diploma, 1990
college         float  %9.0g                  % adults college degree, 1990
overweight      float  %9.0g                  % overweight
inactive        float  %9.0g                  % inactive in leisure time
smokeM          float  %9.0g                  % male adults smoking
smokeF          float  %9.0g                  % female adults smoking
smokeT          float  %9.0g                  % adults smoking
motor           float  %9.0g                  Age-adjusted motor-vehicle
                                                related deaths/100000
-------------------------------------------------------------------------------
Sorted by:  state
```

图3.28 画出了四个地区(*region*)在非积极休闲(*inactive*)人口比例上的中位数。我们看到了明显的地区差异：非积极休闲率在南部地区最高(36%，South)，西部地区最低(21%，West)。请注意，纵轴有自动标注的"p 50 of inactive"，即第50百分位数或中位数。选项 **blabel(bar)** 要求标注直方条的高度(20.9 等)。**bar(1, bcolor(gs(10))** 设定第一个提到的 *y* 变量对应的条形应当填充中度浅灰色。

```
. graph bar (median) inactive, over(region) blabel(bar)
      bar(1, bcolor(gs10))
```

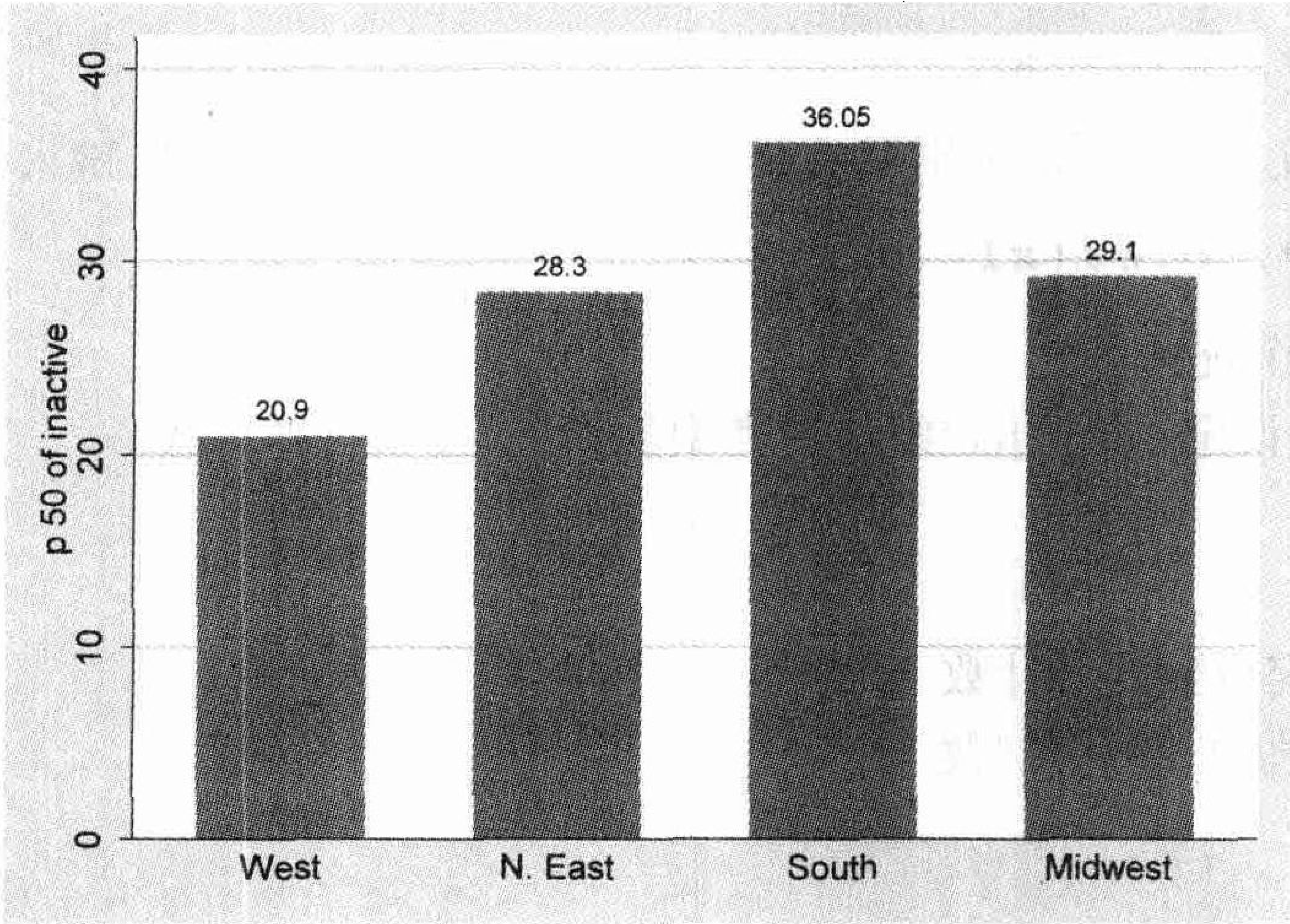


图3.28

通过增加另一个变量 *overweight*(超重)，并将其对应的直方条填充更深灰色，图3.29 进一步体现了这一作图方法。图3.29 中的直方条标签要用中等字号，即 **size(medium)**，这使得它们比图3.28 中默认设定 **size(small)** 的字号更大。**size( )** 子选项的其他可能选择还包括 **tiny**(极小)、**medsmall**(中小)或 **large**(大)等字号标签。对于完整的变量清单，请见 **help textsizestyle**。图3.29 显示身体超重流行程度的地区差异没有非积极休闲率的差异明显，尽管这两个变量的中位数都是在南部和中

西部最高。

```
. graph bar (median) inactive overweight, over(region)
      blabel(bar, size(medium))
      bar(1, bcolor(gs10)) bar(2, bcolor(gs7))
```

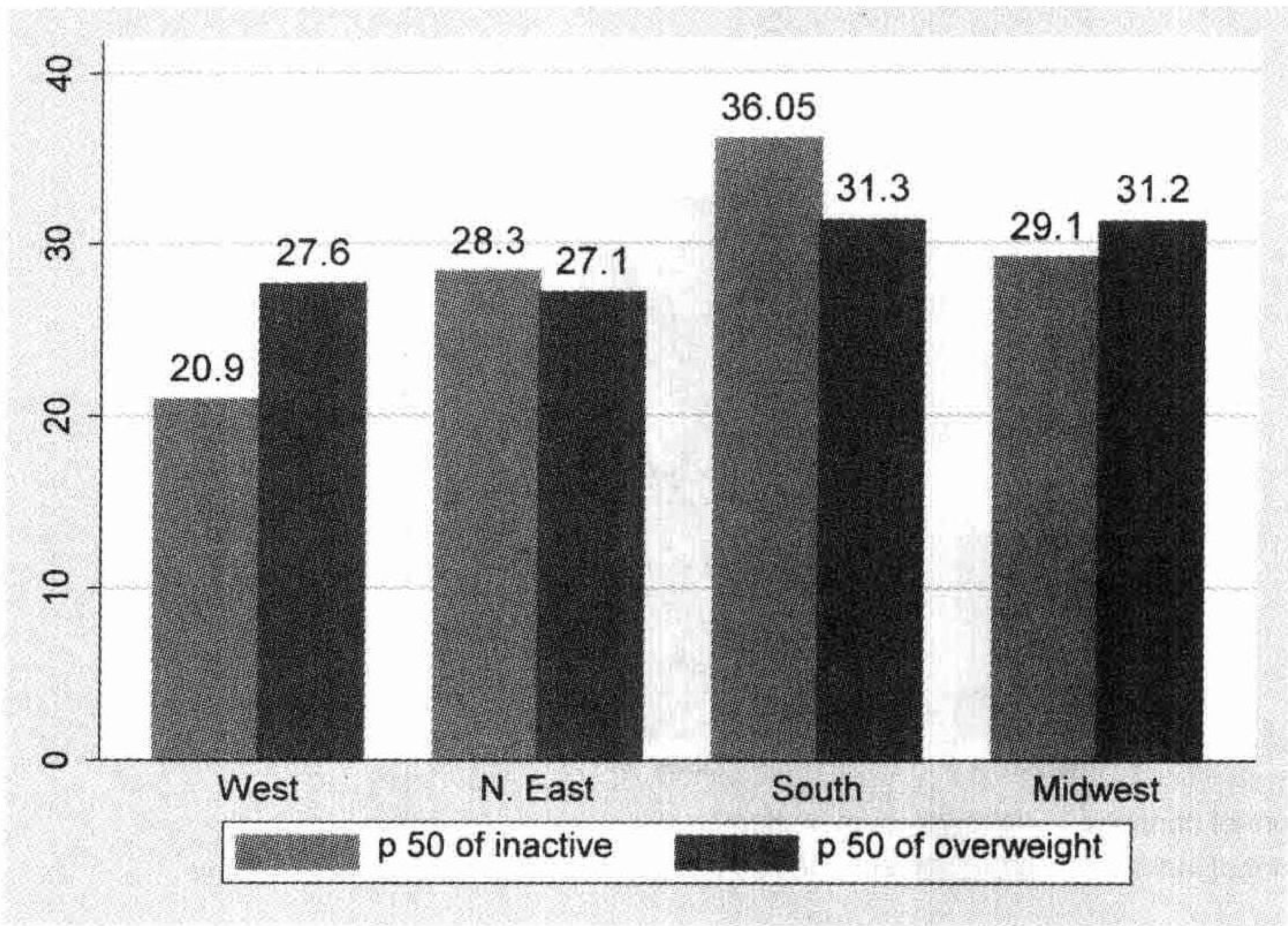


图 3.29

数据 *statehealth.dta* 中的风险指标包括每十万人口中的车祸死亡人数(*motor*)。下面的图 3.30 先按地区将其进行分组,然后在地区内再按低收入和高收入(州户收入中位数是低于还是高于全国中位数)将各州进一步划分成不同下属分组,最后揭示车祸数量与财富存在着惊人的相关。在每一地区内部,低收入的州呈现出更高的平均车祸死亡率。要是按同一收入类别之间的比较,车祸死亡率在南部地区较高,而在东北部地区较低。命令中两个 **over** 选项的顺序控制了它们在组织图形中的顺序。对于本例而言,我们选择了水平条形图(horizontal bar chart)或 **hbar** 。在这种水平条形图中,**ytitle** 和 **yline** 等选项所指的都是横轴。**yline**(**17.2**)标示了总平均数水平。

```
. graph hbar (mean) motor, over(income2) over(region) yline(17.2)
      ytitle("Mean motor-vehicle related fatalities/100000")
```

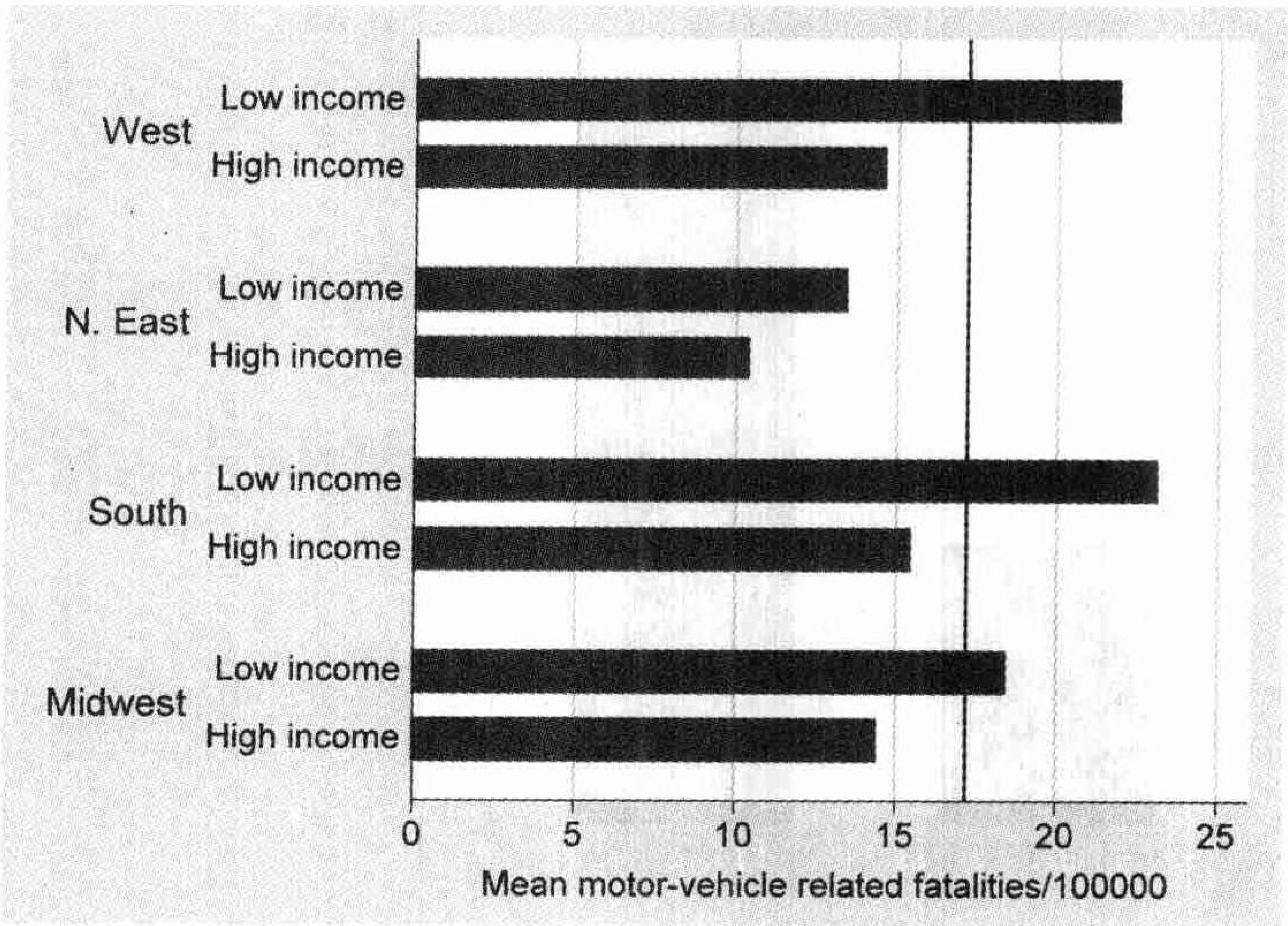


图 3.30

如图3.31所示，直方条也可以层叠(stacked)在一起。这个图是根据阿拉斯加的民族数据(*AKethhnic.dta*)画出的，该图应用所有的默认设定按社区类型(村、镇、城市)来展示人口的民族构成。

```
. graph bar (sum) nonnativ aleut indian eskimo, over(comtyp) stack
```

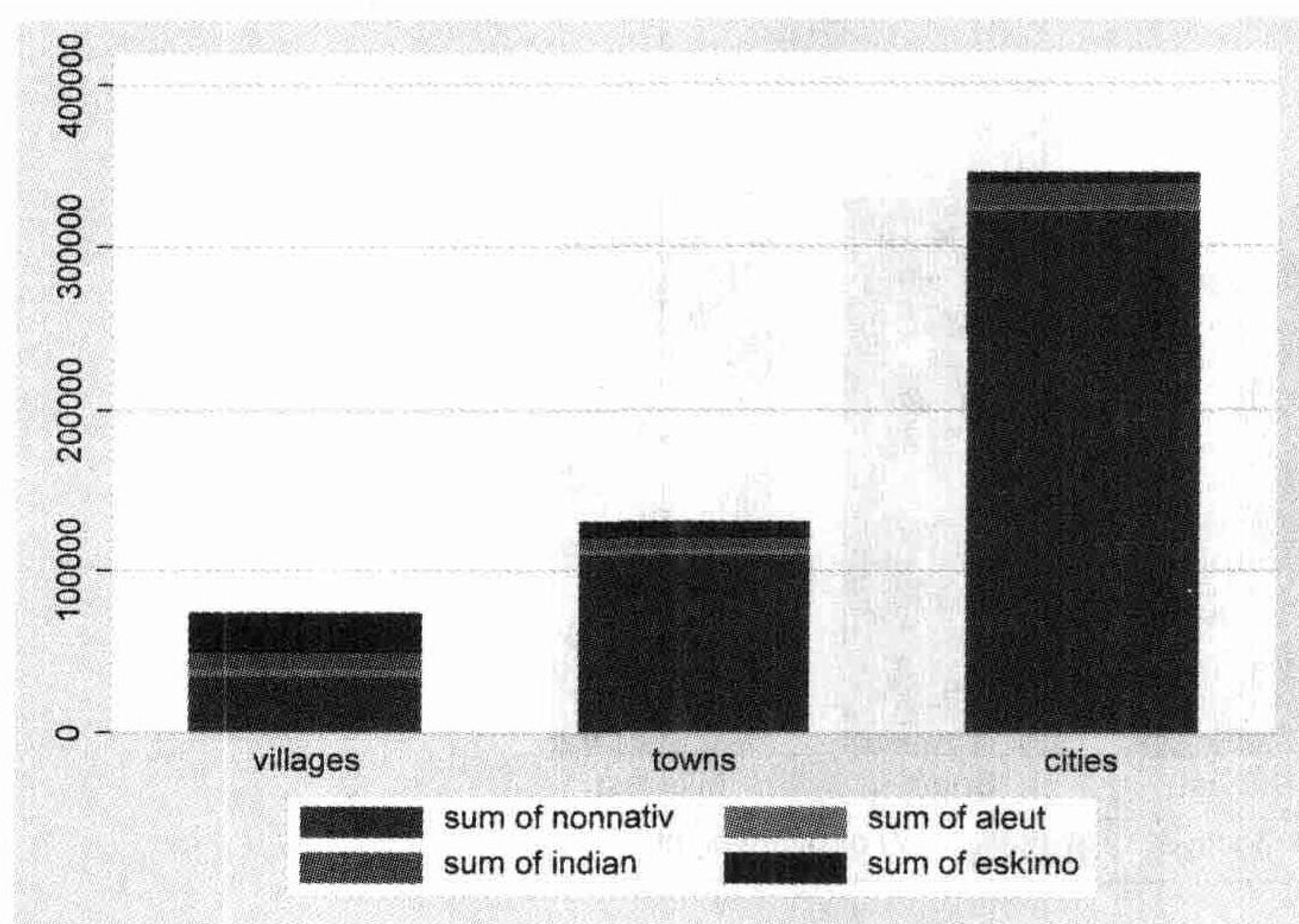


图3.31

图3.32通过改进图例(legend)和坐标轴标签(axis labels)重画了该图。**over**选项现在包括了子选项(suboptions)，这些子选项重新添加社区类型标签，因此在横轴上提供了更多的信息。**legend**选项定义了与直方条层叠的顺序相同的四行图例，并将其放置在图中11点钟的空白位置上。它还改进了图例标签。而选项**ytitle**、**ylabel**以及**ytick**则设定了纵轴的格式。

```
. graph bar (sum) nonnativ aleut indian eskimo,
      over(comtyp, relabel(1 "Villages <1000" 2 "Towns 1000~10000"
         3 "Cities >10000"))
      legend(rows(4) order(4 3 2 1) position(11) ring(0)
         label(1  "Non-native") label(2 "Aleut")
         label(3 "Indian") label(4 "Eskimo"))
      stack ytitle(Population)
      ylabel(0(100000)300000) ytick(50000(100000)350000)
```

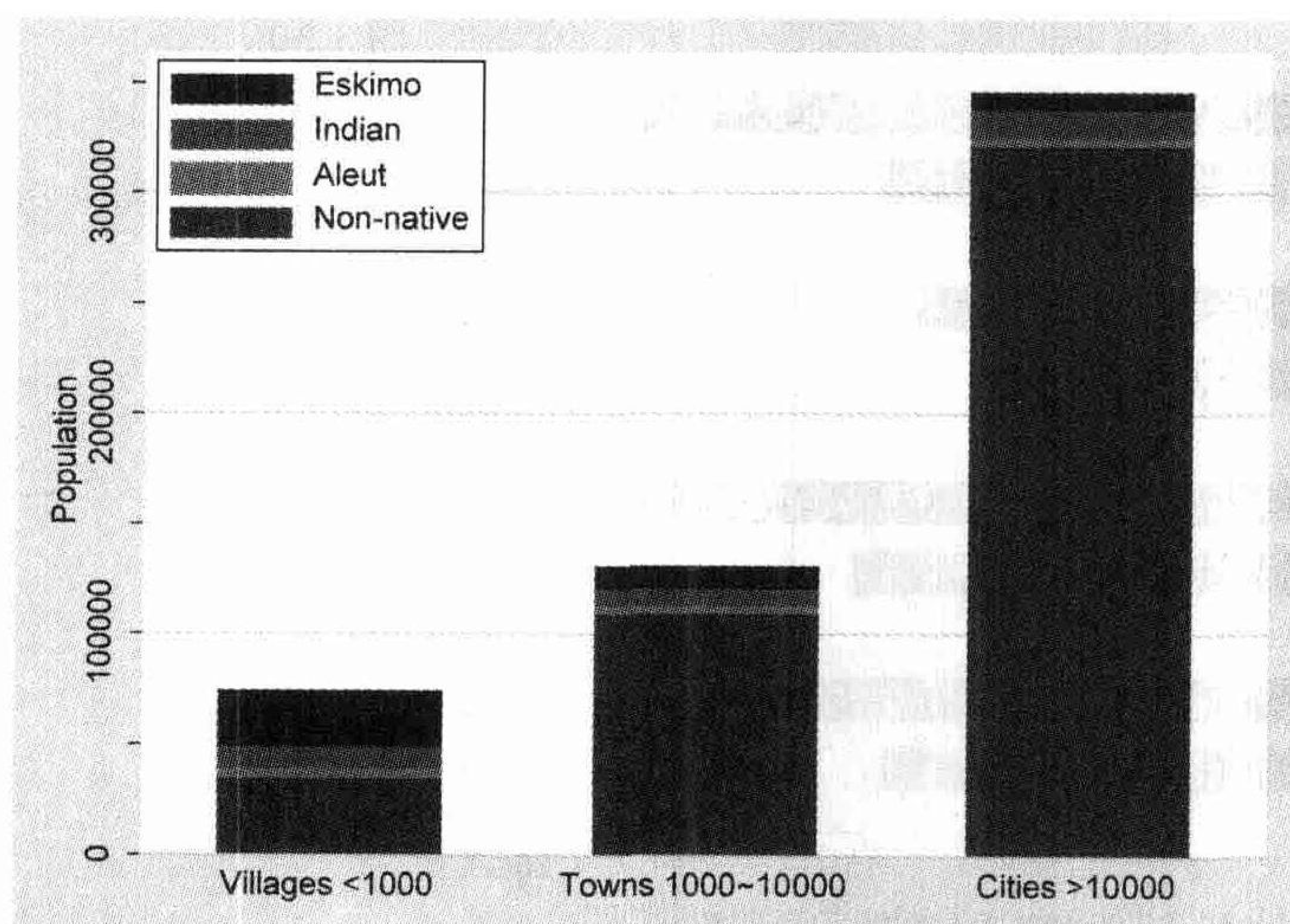


图3.32

图 3.32 针对与图 3.27 中饼图相同的变量进行制图，但是它们的显示却完全不同。尽管饼图显示了每种社区类型内部民族的相对规模（百分比），但是条形图却显示的是它们的绝对规模。因此，图 3.32 告诉我们一些图 3.27 所不能反映的内容：阿拉斯加爱斯基摩人口中的大部分都居住在乡村。

点 图

点图（dot plot）所服务的目的与条形图极为相同：直观地比较一个或更多测量型变量的概要统计。两类图形的组织和 Stata 选项大都类似，包括概要统计的选择在内。要想看到变量 *x*、*y*、*z* 和 *w* 在中位数上比较的点图，键入：

```
. graph dot (median) x y z w
```

要想比较在 *x* 不同类别上 *y* 的平均数比较的点图，键入

```
. graph dot (mean) y, over(x)
```

图 3.33 显示了数据 *statehealth.dta* 中按地区（*region*）划分的男性和女性吸烟率的一幅点图。**over** 选项包含了一个子选项 **sort**（***smokeM***），要求将地区按照其男性吸烟率 *smokeM* 的平均数的值从低到高进行排序。我们还设定用实心三角作为 *smokeM* 的标志记号、用空心圆圈作为 *somkeF* 的标志记号。

```
. graph dot (mean) smokeM smokeF, over(region, sort(smokeM))
      marker(1, msymbol(T)) marker(2, msymbol(Oh))
```

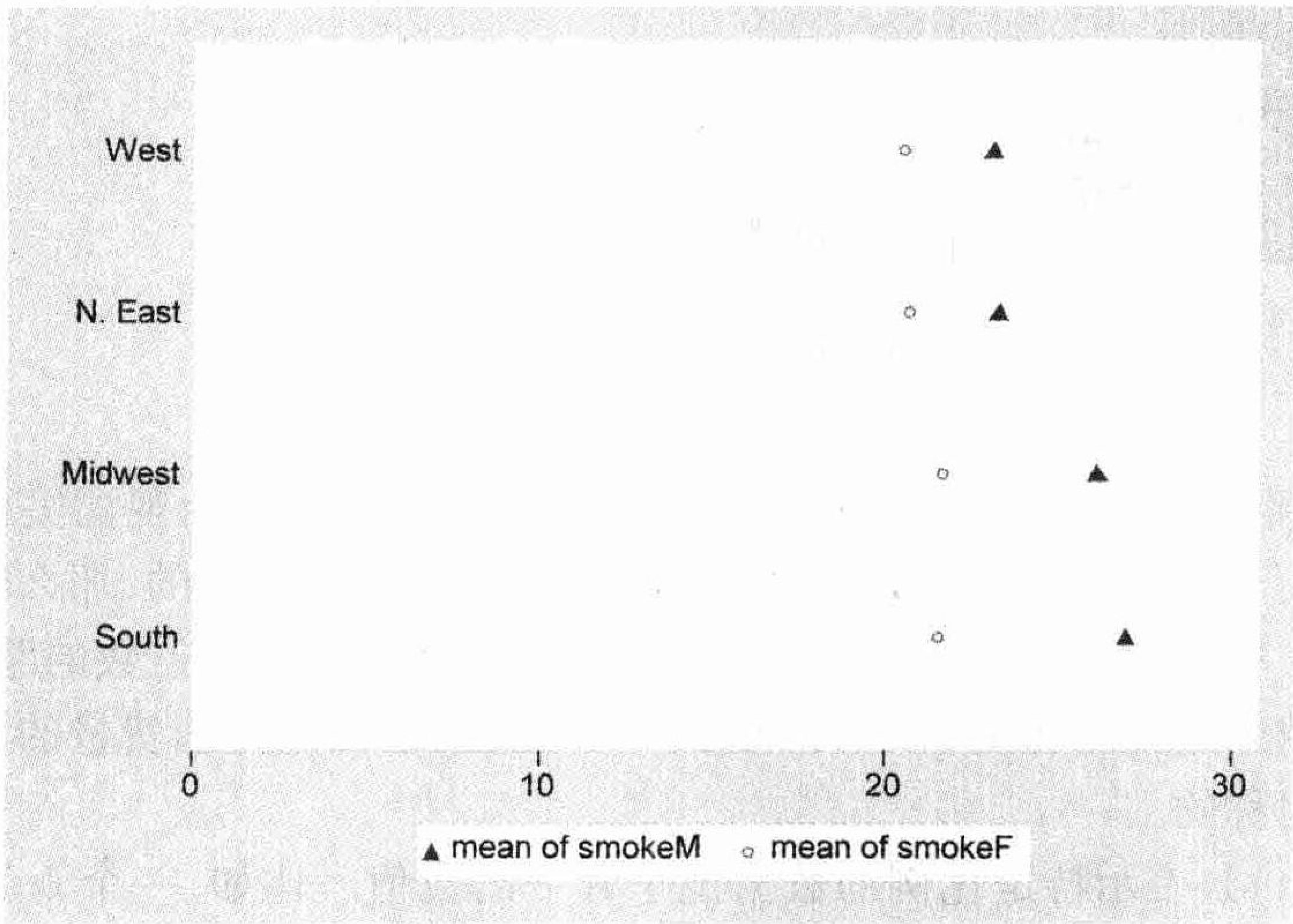


图 3.33

尽管图 3.33 只显示了八个平均数，但是它这样显示在某种程度上便利于若干种比较。我们看到：男性的吸烟率通常更高；就两性而言，都是南部和中西部的吸烟率更高；男性吸烟率的地区差异相对更大。条形图能够传达同样的信息，但是点图的一个优点就在于它的简洁。点图（尤其是将行按所关注的统计量进行排序后，如图 3.33 中那样）即使是在具有十几行或更多行的情况下仍然很容易读。

对称图和分位数图

箱线图、条形图和点图概要描述了测量变量的分布，通过隐藏个体数据点来阐明整体模式。但是，对称图(symmetry plot)和分位数标绘图(quantile plot)则是在一个分布中包含了每个观测案例的数据点。它们比概要图形更难读，但是传递出更具体的信息。

图3.34 呈现了美国50 个州的人均能源消费量 *energy*(取自数据 *states.dta*)的直方图。该分布包含了少数极高消费量的州，这些州恰好都产石油。一条叠加的正态(高斯)曲线表明 *energy* 的左尾小于正态分布，而右尾却大于正态分布，因而符合正偏态的定义。

```
. histogram energy, start(100) width(100) xlabel(0(100)1000)
      frequency norm
```

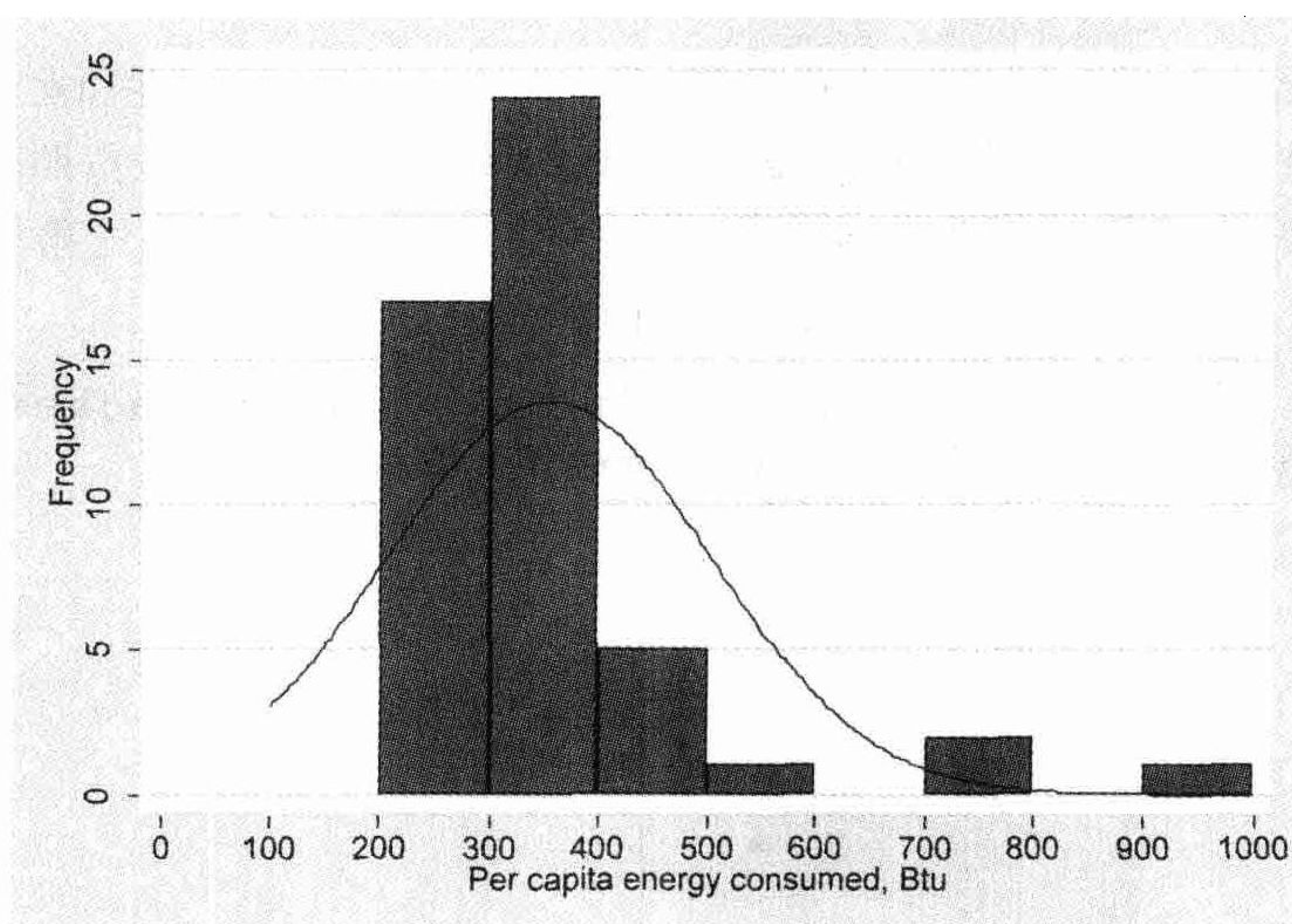


图3.34

图3.35 以对称图的形式描述了这一分布。它以第 *i* 个高于中位数的观测案例的(垂直)距离对第 *i* 个低于中位数的观测案例的距离制图。如果分布是对称的，那么所有的点都将位于对角线上。相反，我们看到的却是正偏态迹象：相对于低于中位数的距离，高于中位数的距离变得越来越大。和图3.34 不同，图3.35 还揭示出能源消费量的分布在中心位置周围是大体对称的。

分位数(quantiles)是表示特定比例的数据位于其下的数值。比如，一个0.3 分位数表明它的值高于30% 的数据值。如果我们将 *n* 个观测案例按升序排列，那么第 *i* 个值就构成了(*i*-0.5)/*n* 分位数。以下命令将计算变量 *energy* 的分位数：

```
. drop if energy >= .
. sort energy
. generate quant = (_n - .5)/_N
```

正如第2 章中提到的，_n 和_N 都是 Stata 的系统变量，只要内存中有数据，它们总是静悄悄地存在。_n 代表当前所在的观测案例序数，而_N 则是观测案例的总数。

分位数标绘图自动计算位于每一数据值以下的观测案例比例有多少，并像图3.36

中那样以图形方式显示这些结果。分位数标绘图为那些手头没有原始数据的人提供了一种图形参考。根据一个恰当添加标签的分位数标绘图,我们能够估计中位数(0.5 分位数)或四分位数(0.25 和 0.75 分位数)等序次统计量。四分位距(IQR)等于 0.25 和 0.75 这两个分位数取值之间的差。我们也可以看一个分位数标绘图来估计落入给定数值以下的观测案例的比例(fraction)。

```
. symplot energy
```

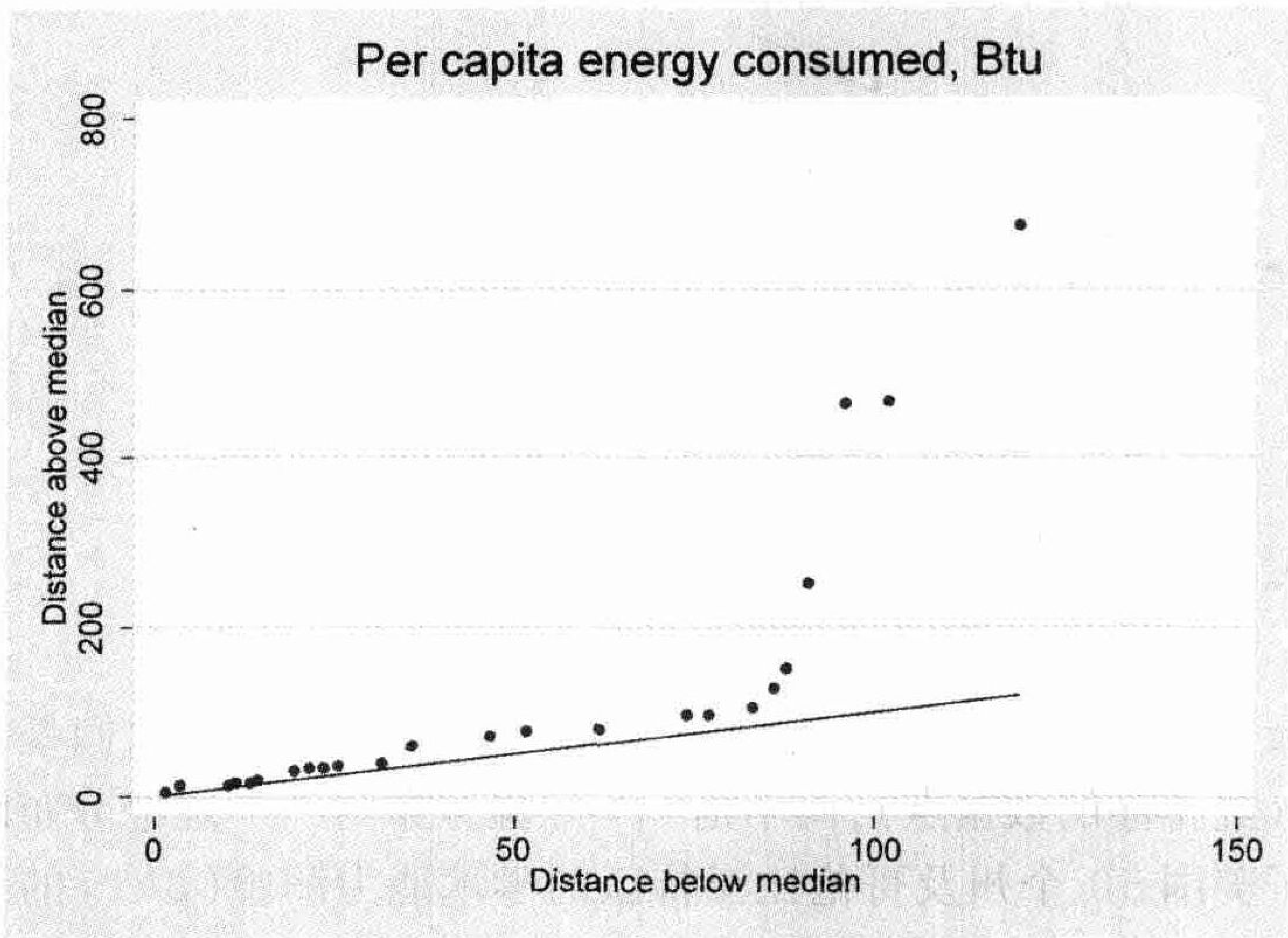


图 3.35

```
. quantile energy
```

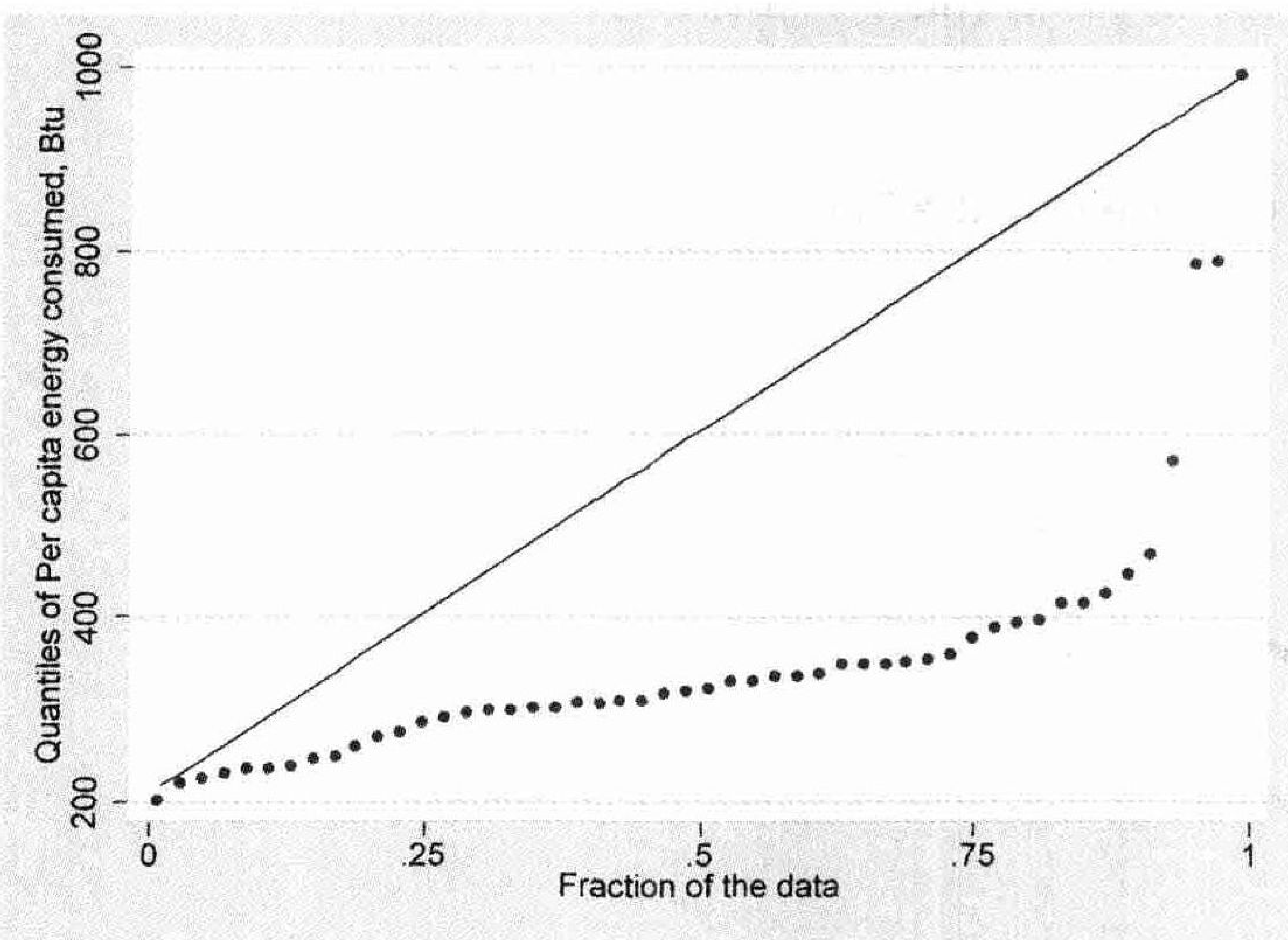


图 3.36

分位—正态图(quantile-normal plot),也被称作正态概率图(normal probability plot),将观测变量分布的分位数与一个具有相同平均数和标准差的理论正态分布的分位数进行比较。这种图可以就变量分布的每个部分对正态性的偏离进行直观的审察,从而有助于指引有关正态性假定的判断和寻找某种正态化转换的方法。图3.37是一幅关于变量 *energy* 的分位—正态图,它确认了我们已经观察到的严重正偏态分布。**grid** 选项要求标注两个分布的 0.05、0.10、0.25(第一四分位数)、0.50(中位数)、0.75(第三四分位数)、0.90 和 0.95 百分位数的坐标刻度。其中,0.05、0.50

和0.95百分位数值显示在顶端和右边的数轴上。

```
. qnorm energy, grid
```

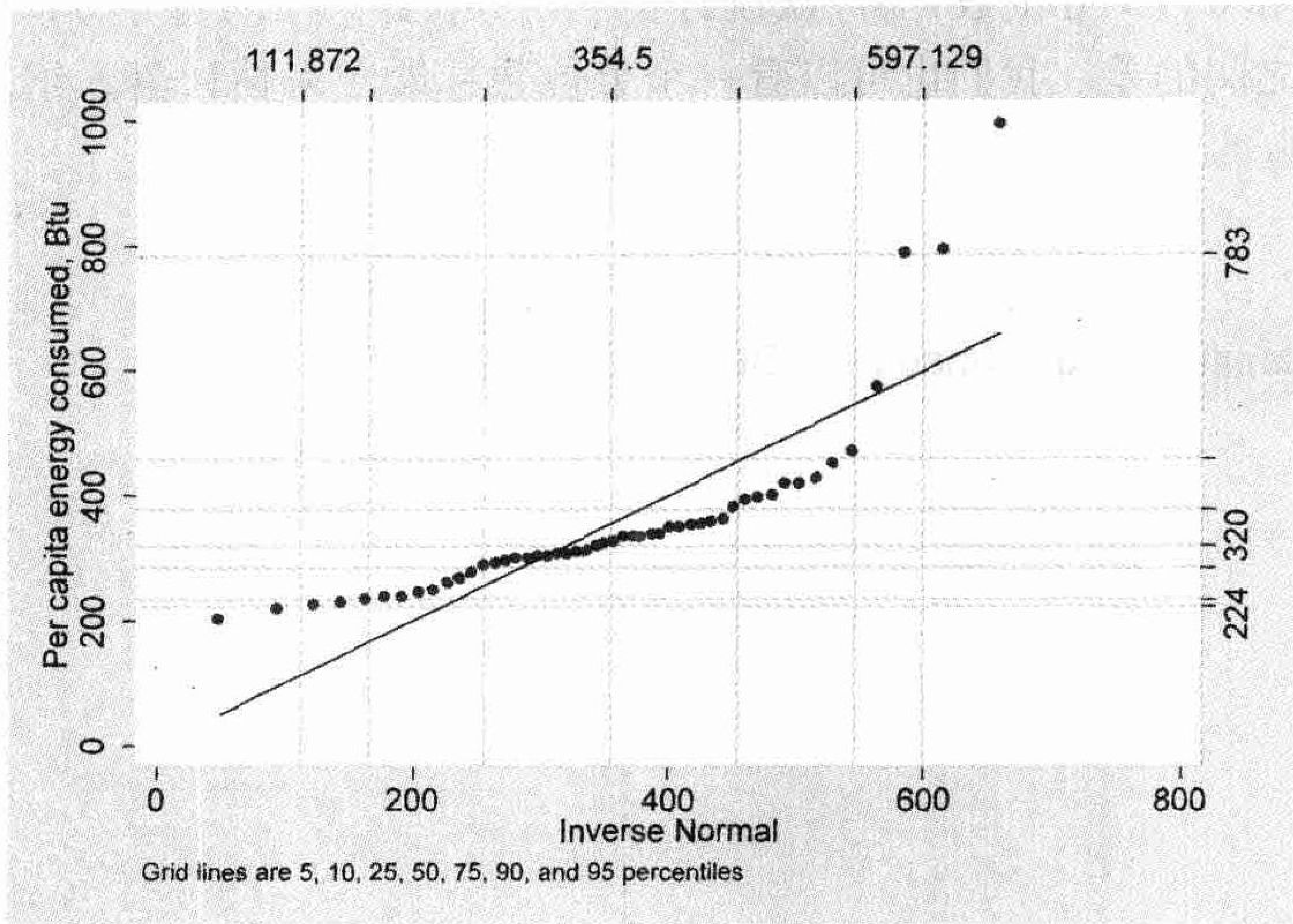


图3.37

分位—分位图(quantile-quantile plot)类似分位—正态图,但是它们是在比较两个经验分布的分位数(经排序的数据点),而不是一个经验分布与一个理论分布的比较。下面的图3.38显示了美国50个州及哥伦比亚特区的学术能力测验(SAT)的平均数学分对平均语言分的分位—分位图。如果两个分布相同,那么我们将看到点都沿着对角线分布。然而,我们看到数据点构成了一条大致平行于对角线的直线,表明这两个变量分布具有不同的平均数、但是具有相似的形状和标准差。

```
. qqplot msat vsat
```

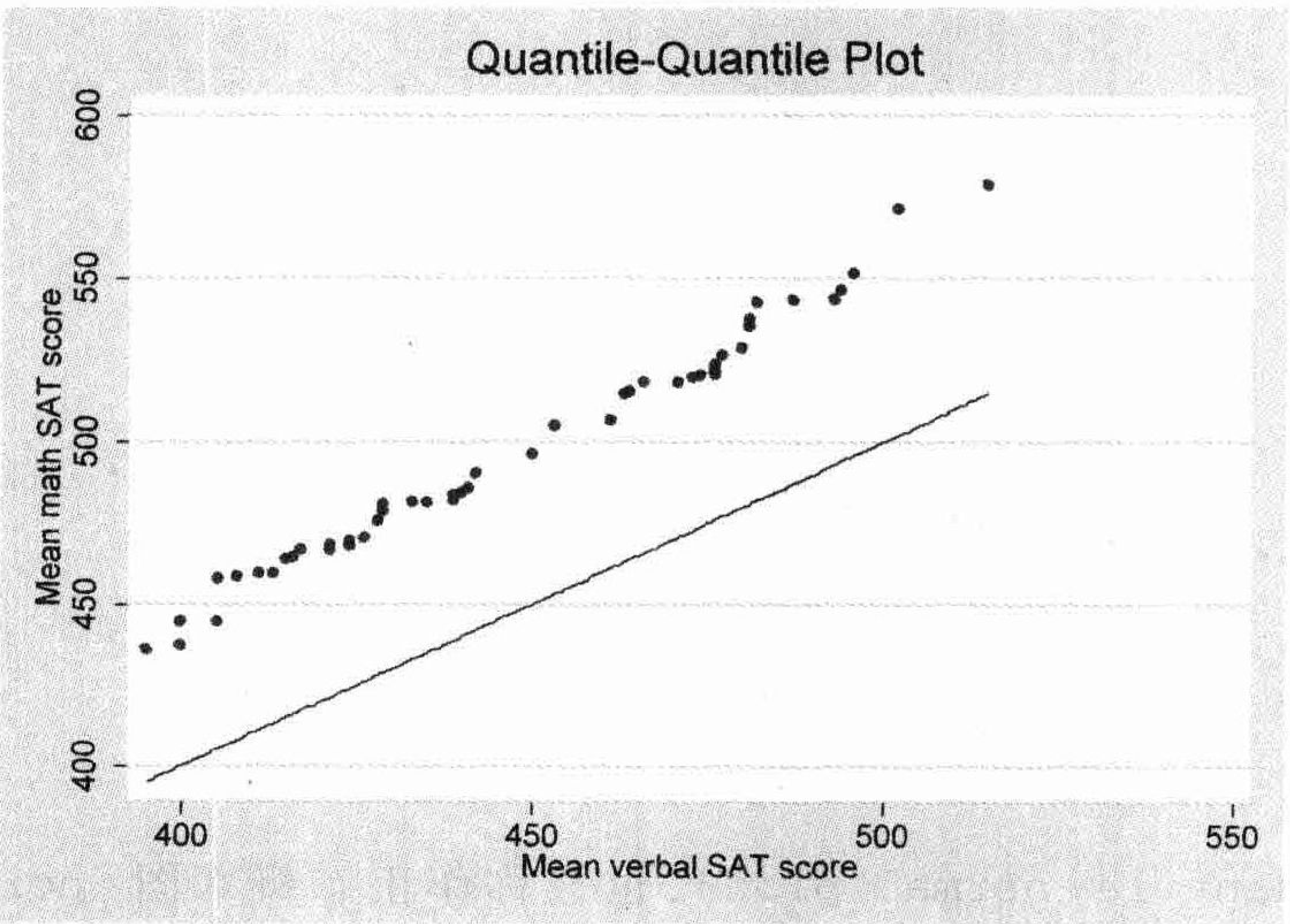


图3.38

《图解回归》(*Regression with Graphics*)(Hamilton,1992a)介绍了如何解读基于分位数的图。Chambers等(1983)提供了更多的内容。相关的Stata命令还包括**pnorm**(标准正态概率图)、**pchi**(卡方概率图)和**qchi**(分位卡方图)。

质量控制图

质量控制图(quality control charts)用于对工业生产等重复过程的产出进行监控。Stata 提供了四种基本类型:c 图、p 图、R 图和 $\bar{x}$ 图。在这些方法被发明之后，又产生了第五种类型，它是由纵向排齐的 $\bar{x}$ 图和 R 图构成，又被称作常规控制图(Shewhart)。Iman(1994)提供了对 R 图和 $\bar{x}$ 图的简要介绍，包括计算控制极限所用的计算表。《基础参考手册》(*Base Reference Manual*)提供了 Stata 命令的细节和公式。这些命令的基本样式如下：

```
. cchart defects unit
```

c 图画出了单位数量(*unit*)中的不一致(nonconformities)或瑕疵(*defects*)数量。基于每个单位出现不一致的数量服从泊松分布的假定，图中以水平线标示上下控制极限。数值超过极限的观测案例被认为是“失控”。

```
. pchart rejects unit ssize
```

p 图画出了单位数量(*unit*)中的不合格产品的比例(即 *rejects* 除以 *ssize*)。上下控制极限由正态分布近似推导出来，计算中考虑了样本规模(*ssize*)的影响。如果不同单位样本规模不同，那么控制极限也将随之变化，除非我们增加一个选项 **stabilize**。

```
. rchart x1 x2 x3 x4 x5, connect(l)
```

用每个样本(sample)从变量 *x1* 到 *x5* 所代表的重复测量来建构一幅 R(即全距，range)图，在本例中指每个案例重复测量 5 次。按样本序数画出每个案例重复测量的全距的图形，并且(此为任选项)以线段连接成连续全距曲线。水平线分别标示平均全距和控制极限。如果这一过程的标准差未知，那么控制极限将根据样本规模进行估计。在标准差 σ 已知时，我们可以在命令中加入这一信息。比如，假设 $\sigma = 10$，可键入：

```
. rchart x1 x2 x3 x4 x5, connect(l) std(10)
```

```
. xchart x1 x2 x3 x4 x5, connect(l)
```

使用变量 *x1* 到 *x5* 的重复测量建构一幅 $\bar{x}$(平均数)图。按样本序数画出每个样本重复测量的平均数，并且以线段连接为平均数曲线。平均数的范围根据样本平均数的总平均数估计得到，控制极限根据样本规模估计得到，除非我们不采用这些默认设定。比如，如果我们知道实际过程具有平均数 $\mu = 50$ 和标准差 $\sigma = 10$，可以用以下方式纳入这些信息：

```
. xchart x1 x2 x3 x4 x5, connect(l) mean(50) std(10)
```

作为替代，我们也可以定义特定的上下控制极限：

```
. xchart x1 x2 x3 x4 x5, connect(l) mean(50) lower(40)
      upper(60)
```

```
. shewhart x1 x2 x3 x4 x5, mean(50) std(10)
```

在一幅图中，垂直排列一幅 $\bar{x}$ 图和一副 R 图。

为示范制作一幅 p 图，我们用数据文件 *quality1 .dta* 中的质量检验数据。

请注意，各单位的样本规模(*ssize*)不同，并且单位(*day*,天)并没有经过排序。

```
Contains data from C:\data\quality1.dta
  obs:            16                          Quality control example 1
 vars:             3                          4 Jul 2005 12:07
 size:           112 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
day             byte   %9.0g                  Day sampled
ssize           byte   %9.0g                  Number of units sampled
rejects         byte   %9.0g                  Number of units rejected
-------------------------------------------------------------------------------
Sorted by:

. list in 1/5

     +------------------------+
     | day   ssize   rejects  |
     |------------------------|
  1. |  58      53        10  |
  2. |   7      53        12  |
  3. |  26      52        12  |
  4. |  21      52        10  |
  5. |   6      51        10  |
     +------------------------+
```

pchart 自动地处理这些复杂情况,所创建的图3.39中的控制极限是变化的。(如果要固定控制极限、而不管样本规模不同的情况,可以加上 **stabilize** 选项)

```
. pchart rejects day ssize
```

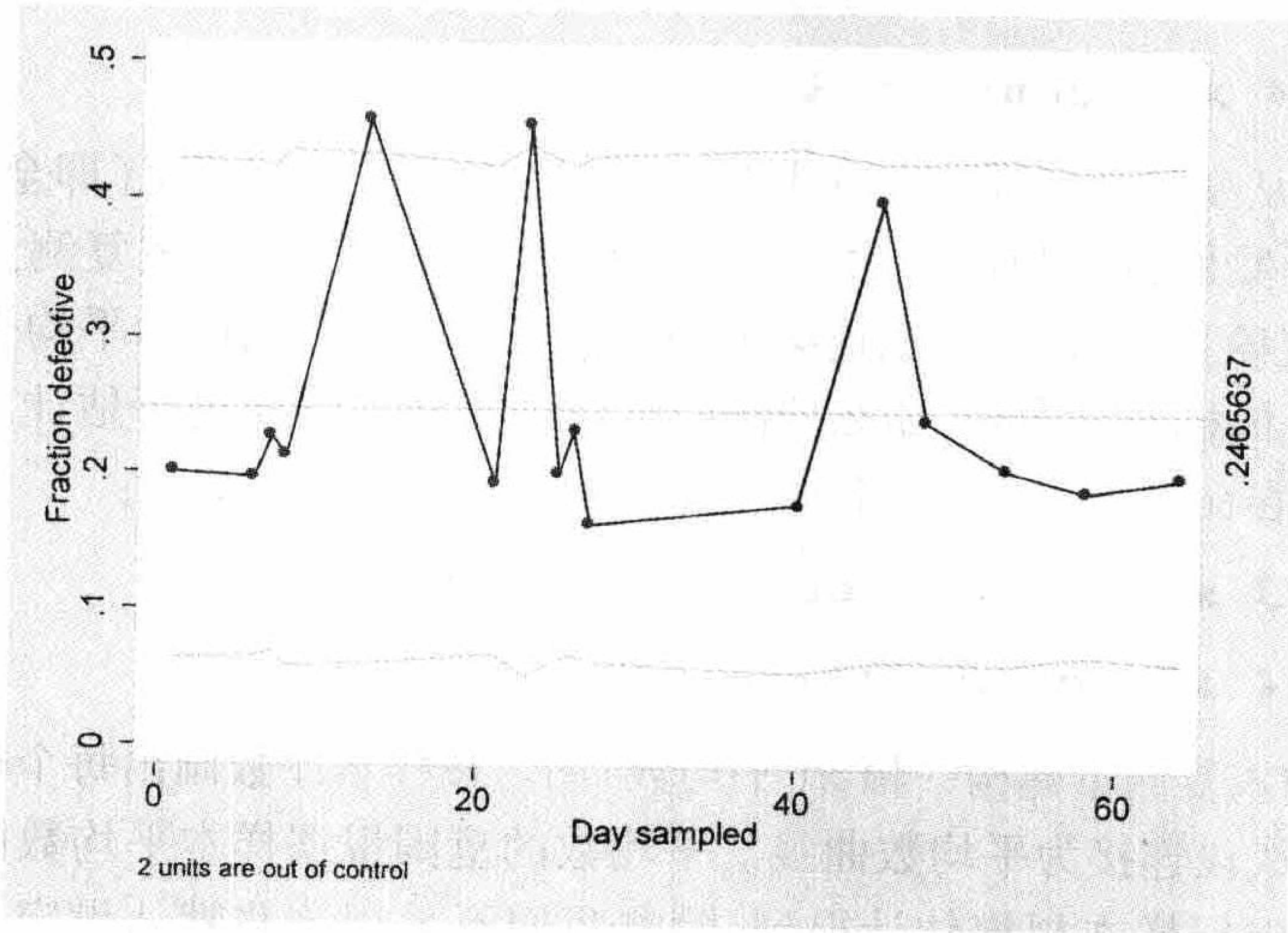


图3.39

借用Iman(1994:662)的数据集*quality2.dta*来示范**rchart**和**xchart**。变量*x1*到*x4*表示来自某个工业生产过程的重复测量;该数据共有25个单位,每个都有4次重复测量。

图3.40的R图画出了这25个单位在全距上的过程变异。**rchart**告诉我们有一个单位的全距“失控”。

图3.41的$\bar{x}$图显示了平均数上的过程变异。这25个平均数中没有一个超出控制极限。

```
Contains data from C:\data\quality2.dta
  obs:            25                          Quality control (Iman 1994:662)
 vars:             4                          4 Jul 2005 12:07
 size:           500 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
x1              float  %9.0g
x2              float  %9.0g
x3              float  %9.0g
x4              float  %9.0g
-------------------------------------------------------------------------------
Sorted by:

. list in 1/5

     +------------------------+
     |  x1     x2     x3    x4 |
     |------------------------|
  1. | 4.6      2      4   3.6 |
  2. | 6.7    3.8    5.1   4.7 |
  3. | 4.6    4.3    4.5   3.9 |
  4. | 4.9      6    4.8   5.7 |
  5. | 7.6    6.9    2.5   4.7 |
     +------------------------+

. rchart x1 x2 x3 x4, connect(l)
```

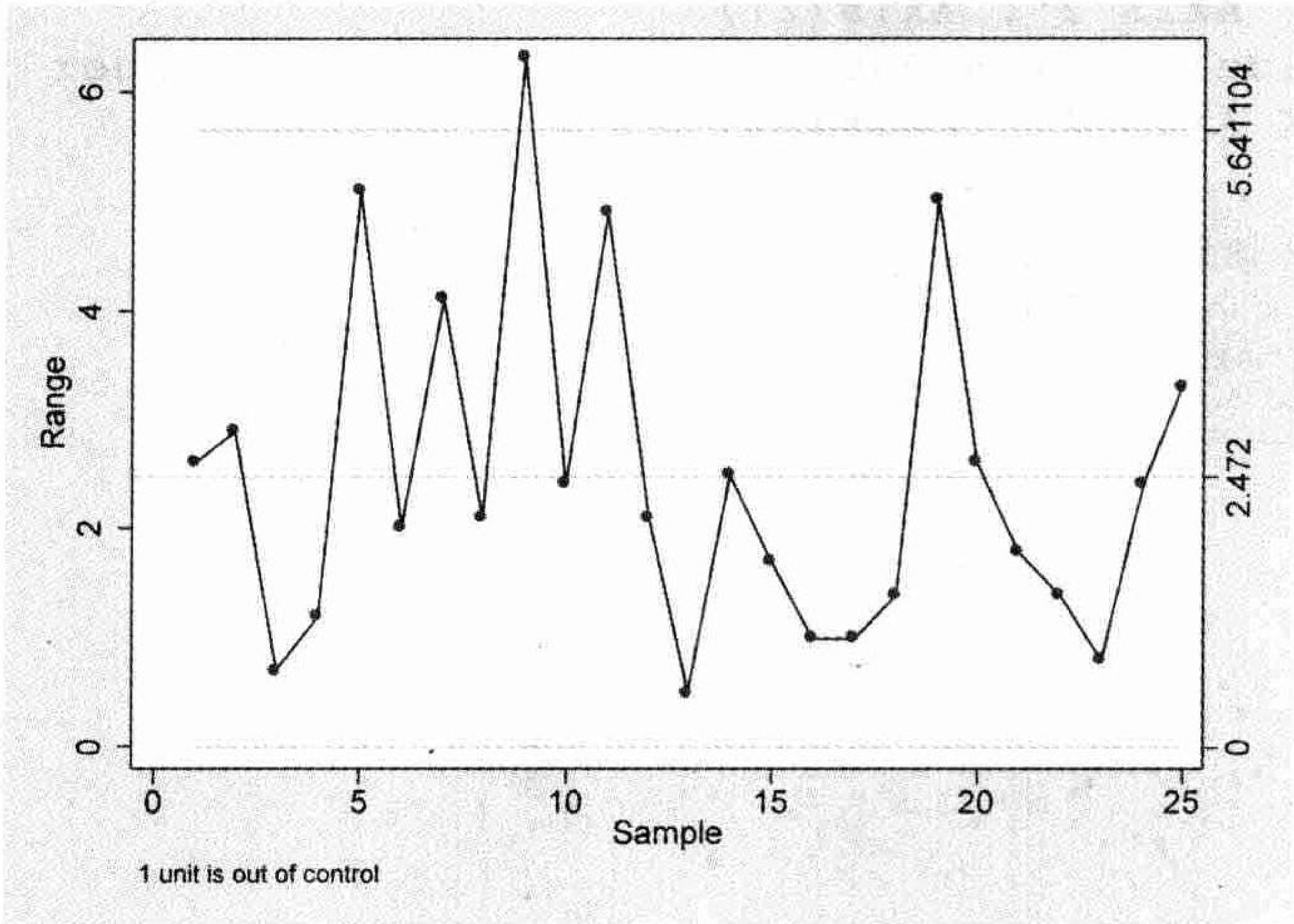


图 3.40

```
. xchart x1 x2 x3 x4, connect(l)
```

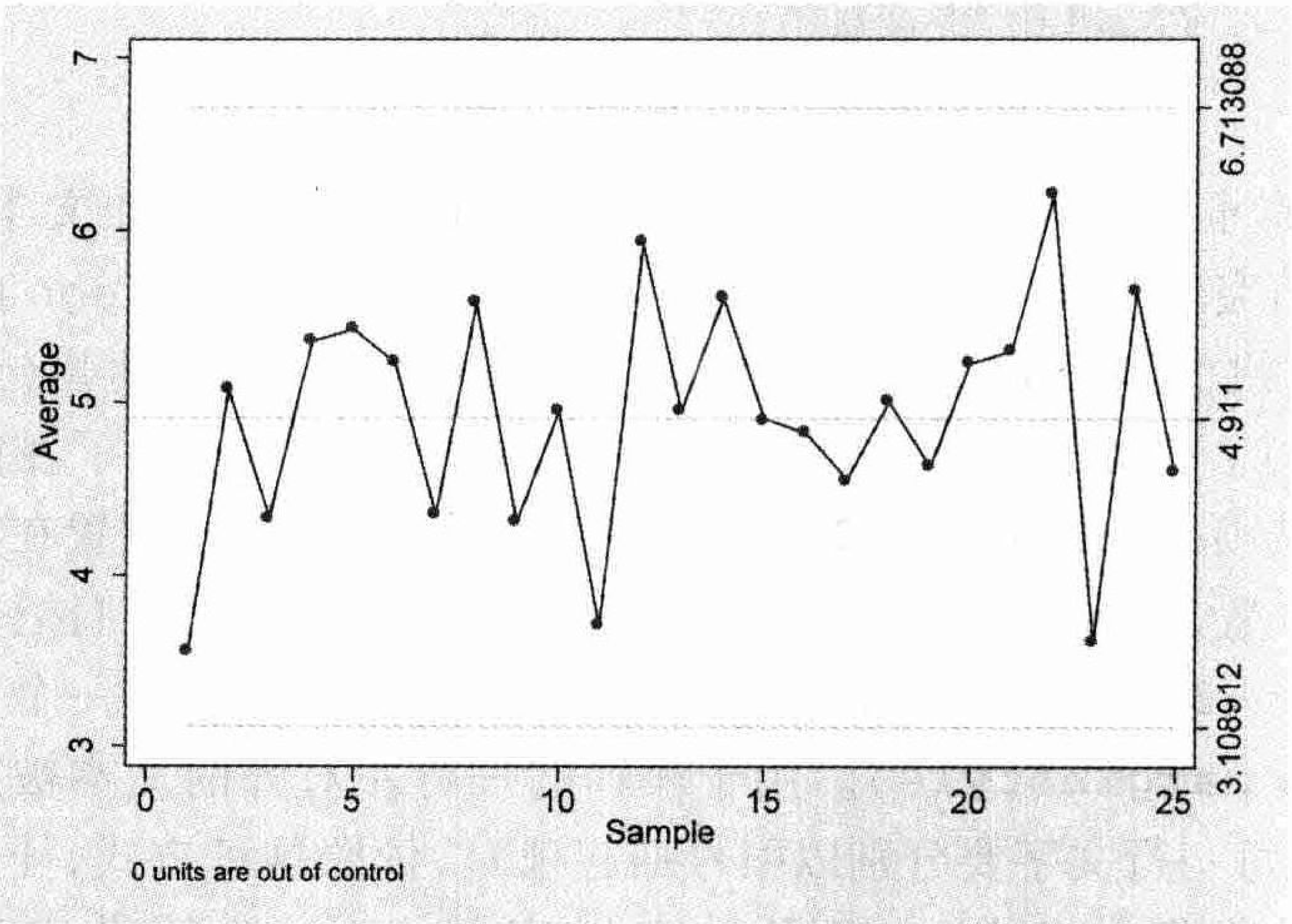


图 3.41

对图形添加文本

可以添加标题(title)、说明(caption)和注(note)以使得图形更具自我解释性。在默认情况下标题和副标题(subtitle)出现在图形的上方;注(比如,说明数据来源)和说明出现在下方。当然,这些默认设定可被忽略。键入 **help title_options** 查看有关标题放置的更多信息,或者用 **help textbox_options** 查看有关它们内容的细节。图3.42 使用数据 *statehealth.dta* 来示范美国各州的吸烟流行率和大学毕业人口的比例的散点图,图中采用以上这四个选项的默认设置。还用选项 **yaxis(1 2)** 设置了图3.42 中左右两侧的 *y* 轴标题,用 **xaxis(1 2)** 设置了顶部和底部 *x* 轴的标题。随后,用选项 **ytitle** 和 **xtitle** 中的子选项 **axis(2)** 又设置了第二个轴标题。正如我们后面将看到的,*y* 轴第二标题不一定非设在右侧,*x* 轴第二标题也不一定非设在上方;但这些都是它们的默认位置。

```
. graph twoway scatter smokeT college, yaxis(1 2) xaxis(1 2)
      title("This is the TITLE") subtitle("This is the SUBTITLE")
      caption("This is the CAPTION") note("This is the NOTE")
      ytitle("Percent adults smoking")
      ytitle("This is Y AXIS 2", axis(2))
      xtitle("Percent adults with Bachelor's degrees or higher")
      xtitle("This is X AXIS 2", axis(2))
```

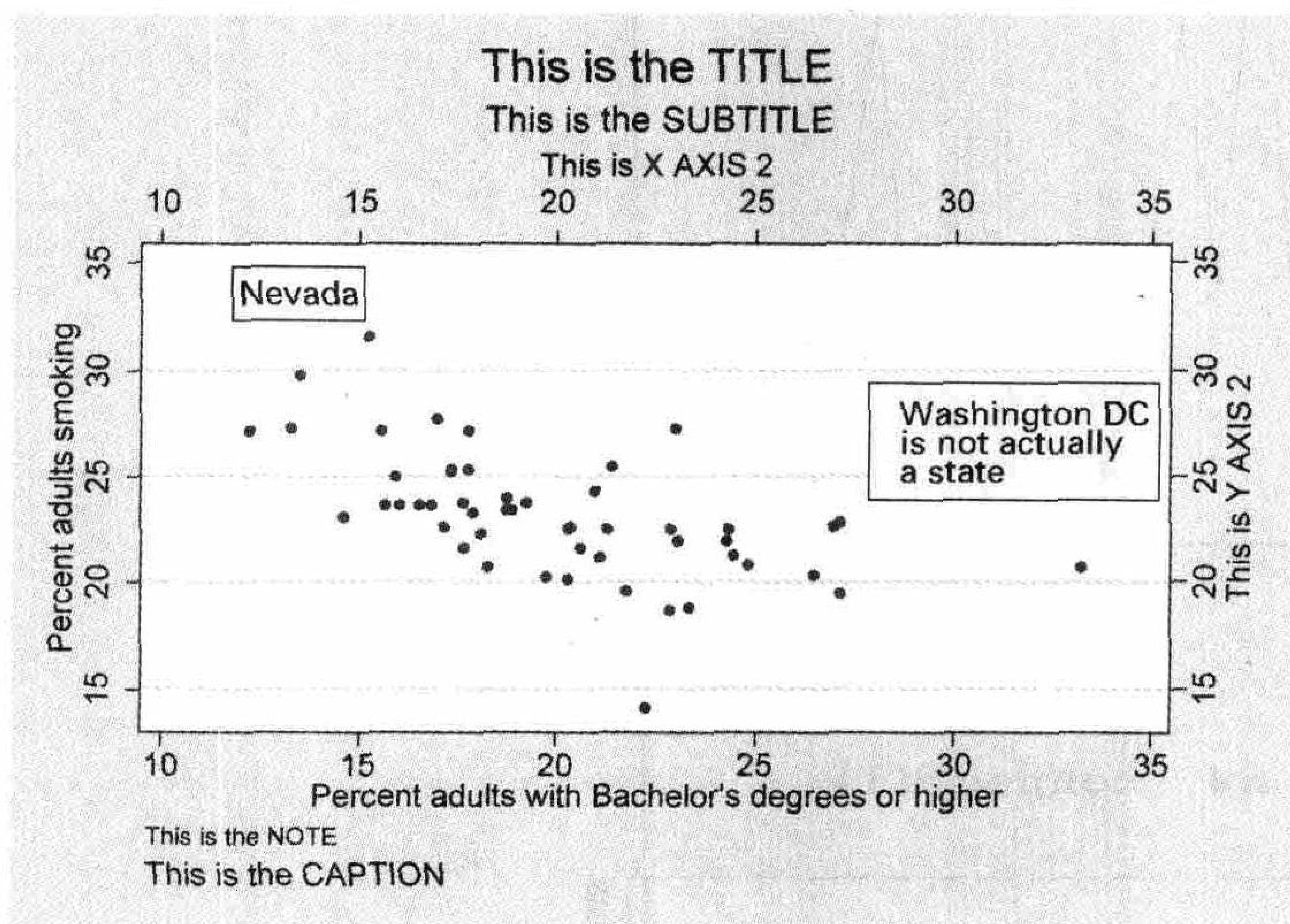


图3.42

标题加在图形之外的文本框内。我们也可以在图形内部的指定位置添加文本框。在这一散点图中有几个特异值突现出来。通过细究,它们是华盛顿特区(*college* 的最大值,位于图中最右边)、犹他州(最低的 *somkeT* 值,位于底部中间)和内华达州(最高的 *smokeT* 值,位于上部左边)。正如图3.43 表明的那样,文本框为我们提供了在图中说明这些观测案例的途径。选项 **text(15.5 22.5"Utah")** 将文字"Utah"放置在散点图中 $y=15.5$、$x=22.5$ 的位置,直接位于犹他州数据点的上方。类似地,我们把文本"Nevada"(内华达州)放在 $y=33.5$、$x=15$ 的位置上,并在该州名字周围画一个方框(带有很小的边距,请见 **help marginstyle**)。用于解释的三行左对齐的文本被放在华盛顿特区数据点附近(其中每一行文字要分别以引号括起来)。依照这种方式,任何文本框或标题都可以有多行;我们分别以一组引号来设定每一行,然后设定对齐

(justification)方式或其他子选项。"Nevada"文本框使用了默认的背景色,然而我们对"Washington DC"(华盛顿特区)文本框则选择了白背景色(请见 **help textbox_options**和 **help colorstyle**)。

```
. graph twoway scatter smokeT college, yaxis(1 2) xaxis(1 2)
      title("This is the TITLE") subtitle("This is the SUBTITLE")
      caption("This is the CAPTION") note("This is the NOTE")
      ytitle("Percent adults smoking")
      ytitle("This is Y AXIS 2", axis(2))
      xtitle("Percent adults with Bachelor's degrees or higher")
      xtitle("This is X AXIS 2", axis(2))
      text(15.5 22.5 "Utah")
      text(33.5 15 "Nevada", box margin(small))
      text(23.5 32 "Washington DC" "is not actually" "a state",
         box justification(left) margin(small) bfcolor(white))
```

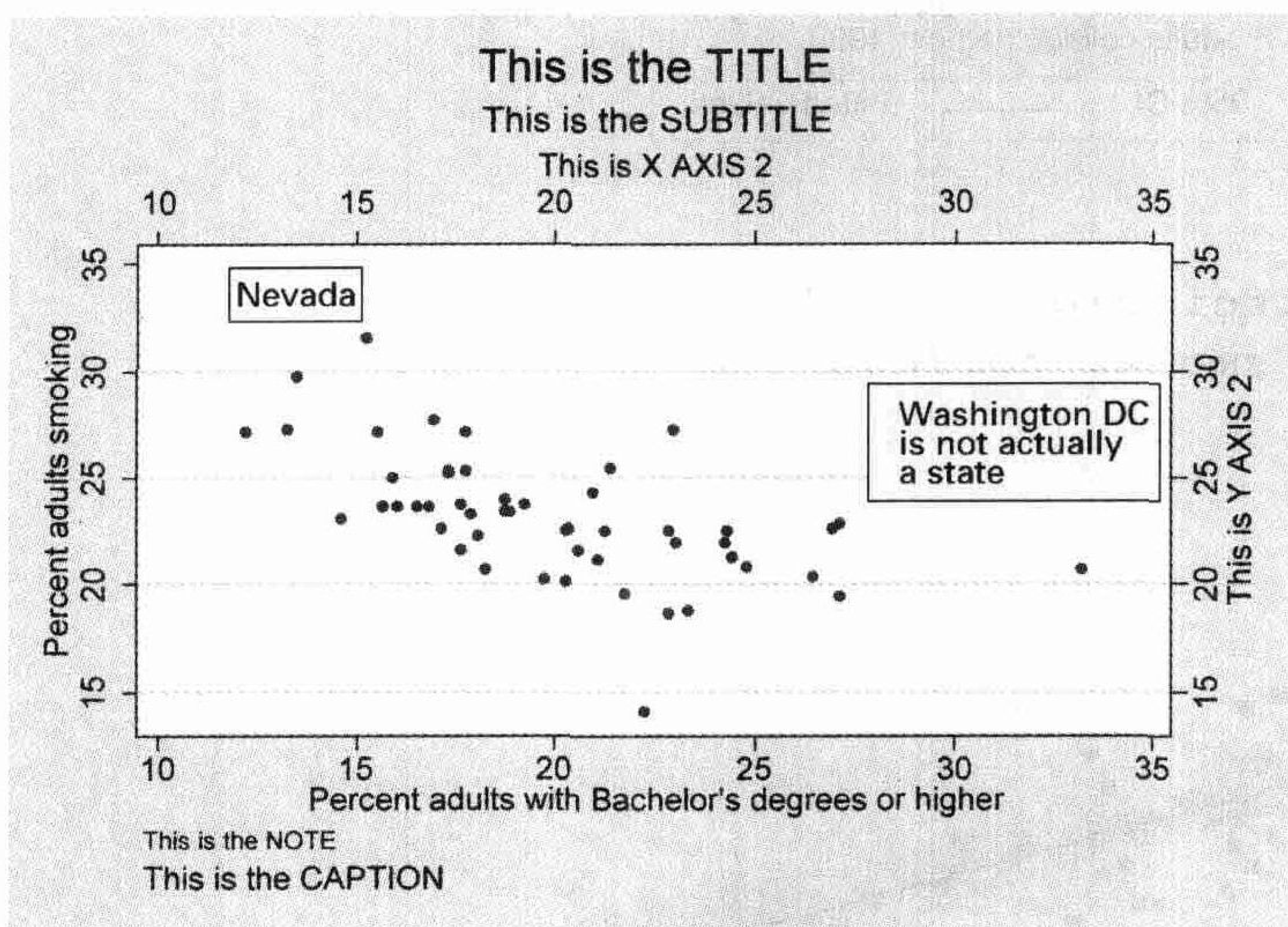


图 3.43

叠并多幅二维图

由功能强大的 **graph twoway** 图族得到的两幅或更多图可以被叠并为一幅统一的图,使其中一幅图置于另一幅图之上。第 1 章的图 1.1 给出了一个简单的例子。**twoway**图族包括诸如 **lfit**(线性回归线)、**qfit**(二次回归曲线)等若干基于模型的类别,以及其他类型。这些图形自身提供了最少量的信息。比如,图 3.44 描述了变量 *smokeT* 对 *college* 的线性回归线(数据为 *statehealth.dta*),图中带有条件平均数及其 95% 置信区间。

正如图 3.45 中所见,当我们将一幅散点图叠并到一幅回归线图之上后会得到信息更丰富的图形结果。为了做到这点,我们实质上是给出两条不同的以"||"分隔的制图命令。

图 3.45 中,第二幅图(散点图)套印第一幅。这一顺序(order)对默认的线条类型(实心、虚线等)有效,也对每一幅子图(sub-plot)所使用的标志记号(方形、圆圈等)有效。更重要的是,它在置信区间上添加了散点,因此这些点仍然是可见的。用户可以试试如果颠倒命令中两幅图的顺序,看看会有什么结果。

```
. graph twoway lfitci smokeT college
```

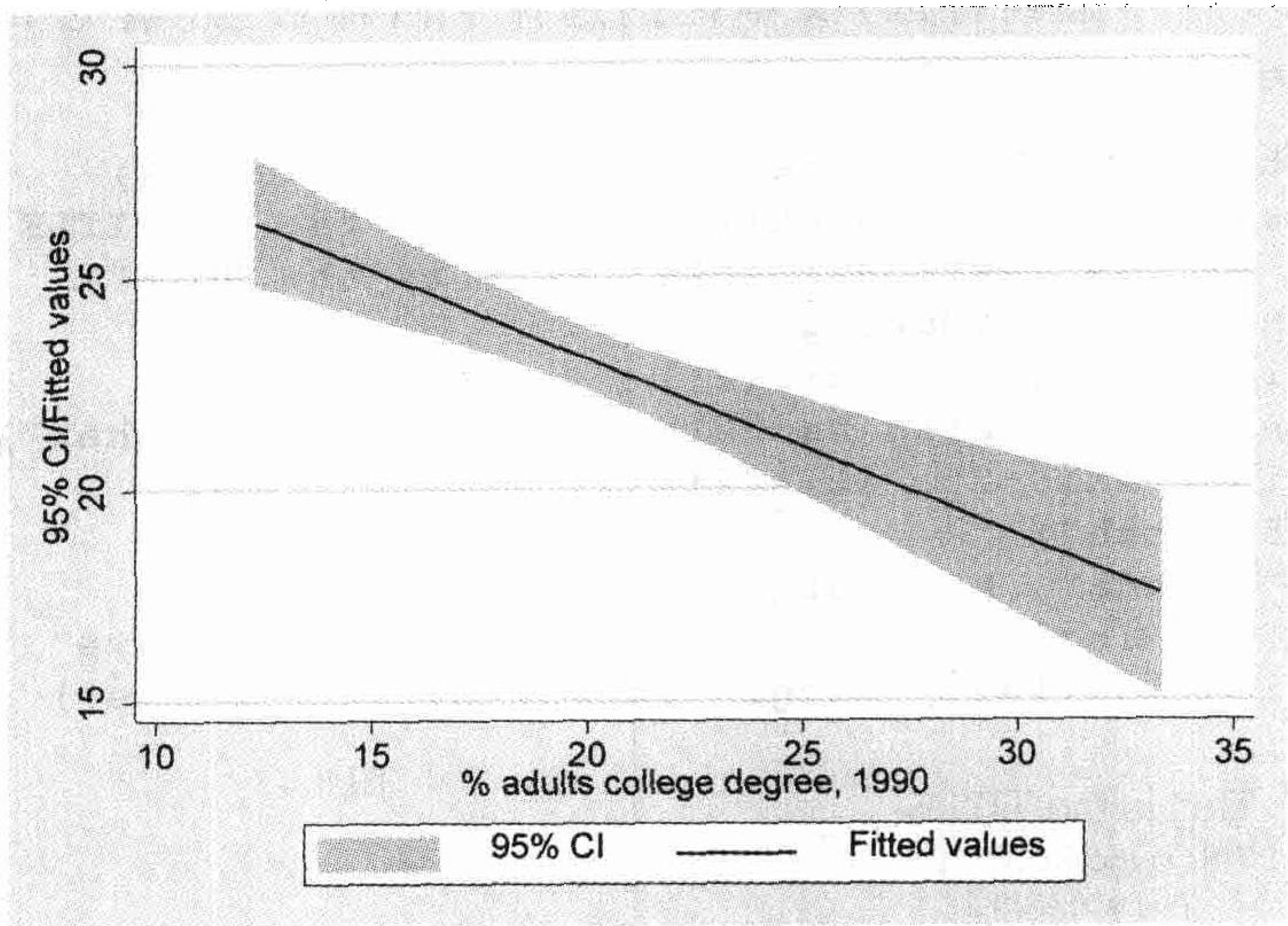


图 3.44

```
. graph twoway lfitci smokeT college
      || scatter smokeT college
```

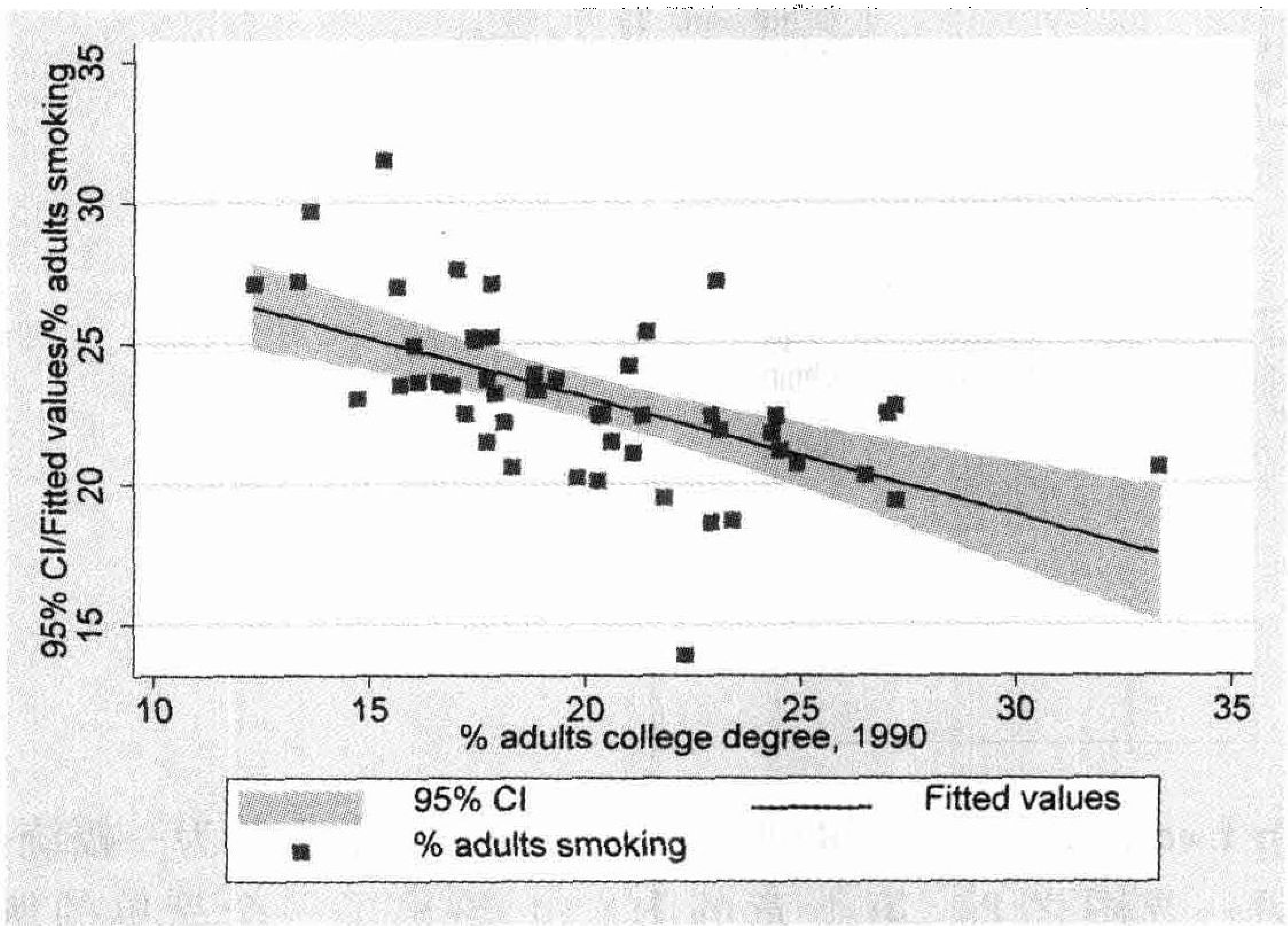


图 3.45

图 3.46 进一步采纳了这一想法，通过选项添加了坐标轴标签和图例对该图形加以改进。因为这些选项作为整体而不只是作为子选项之一应用于图形，因此这些选项被放在第二个分隔符 || 之后的逗号后面。这些选项中的大部分都类似于前面例子中所用到的那些。这里的选项 **order(2 1)** 做了一件新的事情：它忽略三个图例项中的一个，因此只有两个图例项（第 2 项回归线，然后是第 1 项置信区间）出现在图中。将这一图例和图 3.45 加以比较以查看其中的差别。尽管我们在图 3.46 中只列出了两个图例项，但是用 **rows(3)** 设定一个三行的图例格式、仿佛所有三行图例都被保留仍然是必要的。

图 3.46 中叠并在一起的两幅独立图形（**lfitci** 和 **scatter**）共有相同的 *y* 和 *x* 刻度，因此两者应用同一套坐标轴。当关注的变量具有不同的刻度时，我们需要分别标注坐标轴的刻度。图 3.47 以叠并两幅曲线标绘图来举例说明这一点，这个图是根据文件 *gulf.dta* 中的圣劳伦斯湾环境数据制作的。该图将圣劳伦斯湾寒冷的中间层最低

平均水温摄氏度(*cil*)和以平方公里为单位的冬天最大冰面覆盖面积(*maxarea*)两幅时间序列图合并在一起。关于 *cil* 的图使用默认状态下位于左边的第一 *y* 轴,**yaxis(1)**。关于 *maxarea* 的图使用默认状态下位于右边的第二 *y* 轴,**yaxis(2)**。不同的 **ylabel**、**ytitle**、**yline** 和 **yscale** 选项每一个中都包括 **axis(1)** 或 **axis(2)** 子选项,以表明它们所指向的具体 *y* 轴。**ytitle** 引号内的那些空白提供了一种便捷方式将这些标题文字放置在靠近数字标签的地方。(对于不同的方式,请见图 3.48)。包含"Northern Gulf fisheries decline and collapse"的文本框为中等边距;其他选择可见 **help marginstyle**。选项 **yscale(range( ))**将使用经过试验找到的两个时间序列之间的最佳垂直间隔值来定义两个 y 轴的取值范围。

```
. graph twoway lfitci smokeT college
      || scatter smokeT college
      || , xlabel(12(2)34) ylabel(14(2)32, angle(horizontal))
      xtitle("Percent adults with Bachelor's degrees or higher")
      ytitle("Percent adults smoking")
      note("Data from CDC and US Census")
      legend(order(2 1) label(1 "95% c.i.") label(2 "regression line")
         rows(3) position(1) ring(0))
```

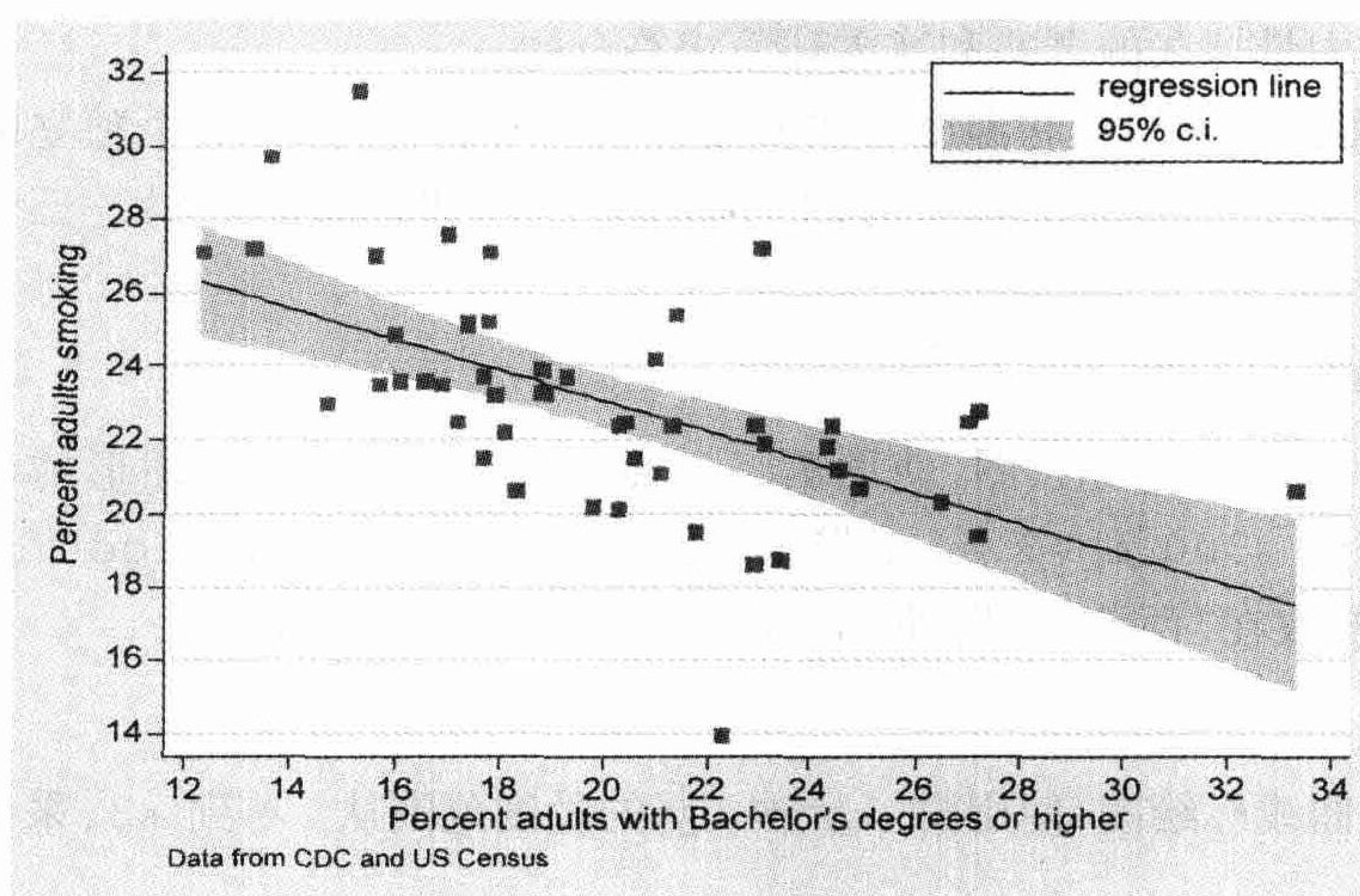


图 3.46

```
. graph twoway line cil winter, yaxis(1) yscale(range(-1,3) axis(1))
     ytitle("Degrees C                              ", axis(1))
     yline(0) ylabel(-1(.5)1.5, axis(1) angle(horizontal) nogrid)
     text(1 1992 "Northern Gulf" "fisheries decline" "and collapse"
        , box margin(medium))
     || line maxarea winter,
     yaxis(2) ylabel(50(50)200, axis(2) angle(horizontal))
     yscale(range(-100,221) axis(2))
     ytitle("                              1000s of km^2", axis(2))
     yline(173.6, axis(2) lpattern(dot))
     || if winter > 1949,
     xtitle("") xlabel(1950(10)2000) xtick(1950(2)2002)
     legend(position(11) ring(0) rows(2) order(2 1)
        label(1 "Max ice area") label(2 "Min CIL temp"))
     note("Source: Hamilton, Haedrich and Duncan (2003); data from
        DFO (2003)")
```

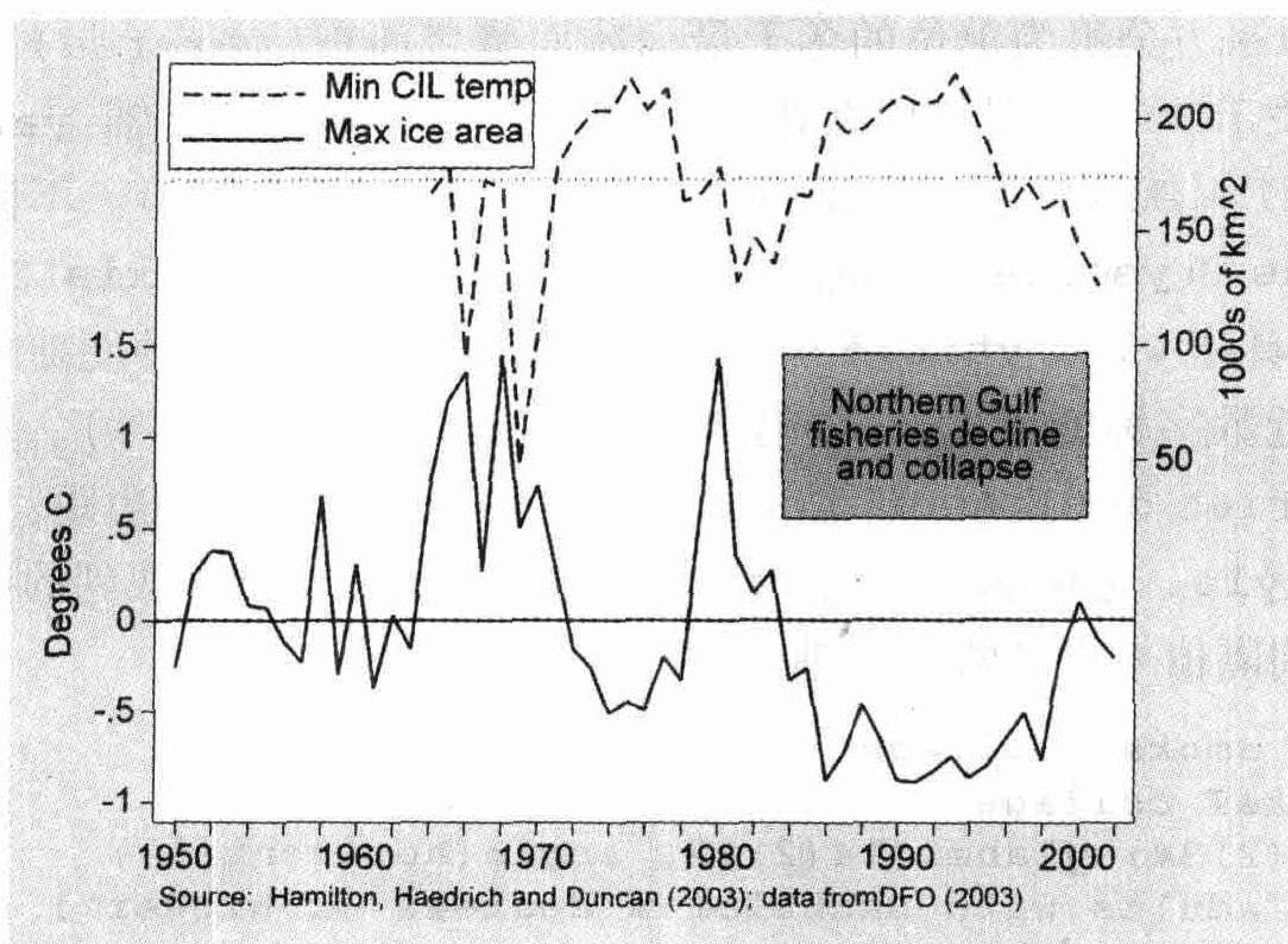


图 3.47

位于图 3.47 中右边的文本框在 1980 年代晚期和 1990 年代早期这段时期处做了标记，包括北部海湾鳕鱼在内的主要渔业在该时期都出现萧条或倒闭。正如该图显示的那样，渔业萧条是和记录中最持久的严寒和冰雪环境相符合的。

为了在同一图中放置鳕鱼捕捞量和温度与冰面覆盖面积，我们就需要三个独立的纵轴刻度。图 3.48 包含三幅图叠并的图形，所有的 *y* 轴都位于左边（默认状态）。三个成分的标绘图的基本形式如下：

connected *maxarea winter*

先将 *maxarea* 对 *winter* 画连线标绘图，使用第三 *y* 轴（将位于我们最后图形的最左边）。*y* 轴的量度范围是从 -300 到 +220，不带水平线栅格。它的标题是“Ice area, 1 000 km^2”。该标题被放在“西北”位置处，采用选项 **placement(nw)**。

line *cil winter*

再用 *cil* 对 *winter* 做曲线标绘图，使用第二 *y* 轴。*y* 的量度范围从 -4 到 +3，采用默认标签。

connected *cod winter*

再用 *cod* 对 *winter* 做连线标绘图，使用第一 *y* 轴。标题位置是“西南”，即

placement(sw)。

将这三个成分图放在一起，图 3.48 对应的完整命令如下所示。对每一个层叠图的 *y* 轴范围都是根据试验找到的三个序列之间“恰当”垂直间隔量而定的。应用于整个图形的选项，还将分析限定在 1959 年以来的年份，并设定了图例和 *x* 轴标签，还要求加上了垂直的栅格线。

使用 Do 文件制图

像图 3.48 这样的复杂图形要求 graph 命令有许多行，每行都很长（尽管 Stata 将整个命令只作为一个逻辑行）。第 2 章中介绍过的 Do 文件有助于编写这类多行的命令。它们也使得易于保存这些命令以备将来再用，也许我们以后还需要对这个图形进行修改或重画该图形。

```
. graph twoway connected maxarea winter, yaxis(3)
    yscale(range(-300,220) axis(3)) ylabel(50(50)200, nogrid axis(3))
    ytitle("Ice area, 1000 km^2", axis(3) placement(nw))
    clpattern(dash)
    || line cil winter, yaxis(2) yscale(range(-4,3) axis(2))
    ylabel(, nogrid axis(2))
    ytitle("CIL temperature, degrees C", axis(2)) clpattern(solid)
    || connected cod winter, yaxis(1) yscale(range(0,200) axis(1))
    ylabel(, nogrid axis(1))
    ytitle("Cod catch, 1000 tons", axis(1) placement(sw))
    || if winter > 1959,
    legend(ring(0) position(7) label(1 "Max ice area")
       label(2 "Min CIL temp") label(3 "Cod catch") rows(3))
    xtitle("") xlabel(1960(5)2000, grid)
```

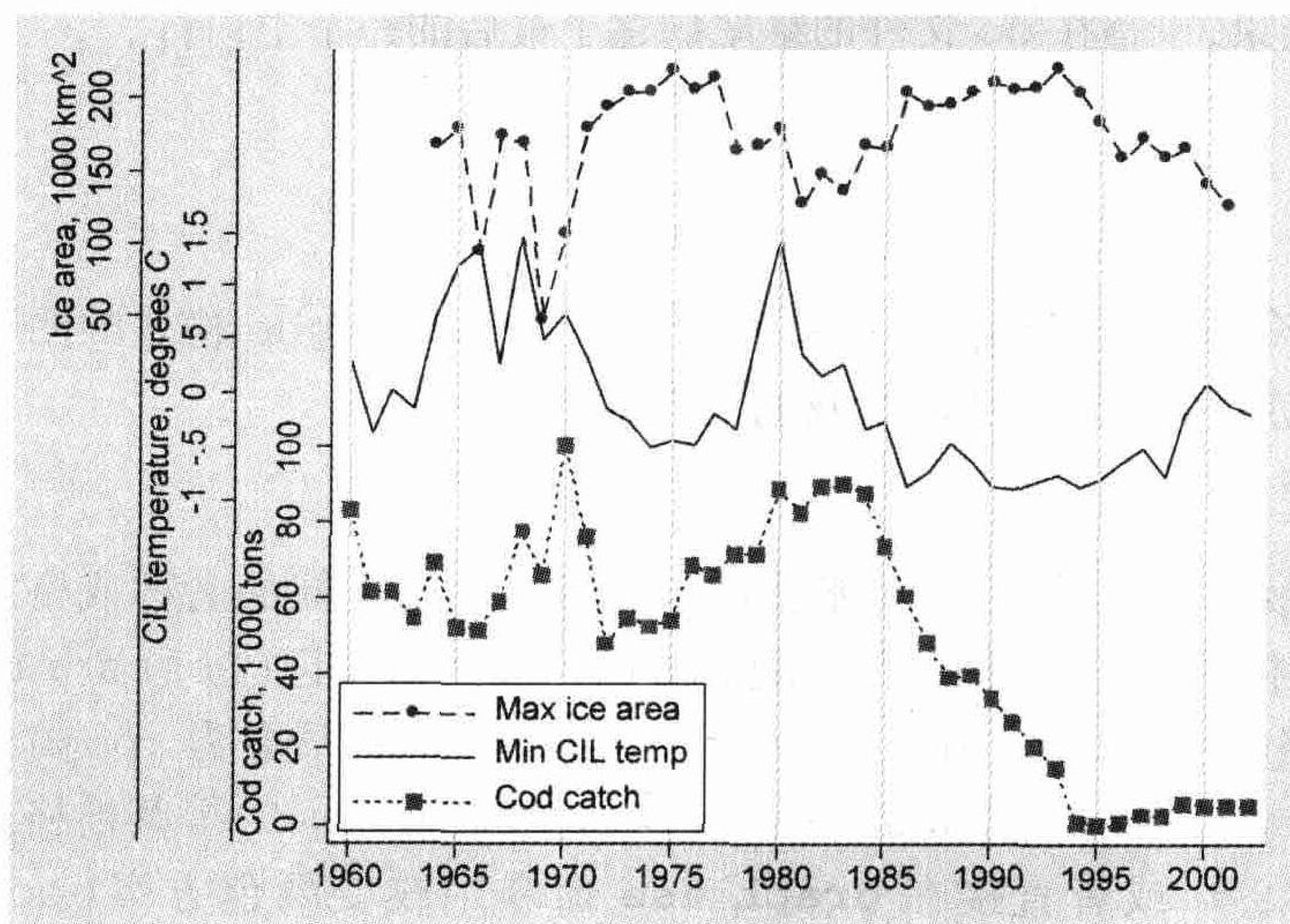


图 3.48

在 Stata 的 Do 文件编辑器中键入后面那段命令，并以文件名 *fig03_48.do* 存盘，它就成为画出图 3.48 的一个新的 do 文件。只要键入：

```
. do fig03_48
```

就会导致该 do 文件得以执行，重新画出这个图形，并且还用两种格式将其加以保存。

```
#delimit ;
use c:\data\gulf.dta, clear ;
graph twoway connected maxarea winter, yaxis(3)
   yscale(range(-300,220) axis(3)) ylabel(50(50)200, nogrid axis(3))
   ytitle("Ice area, 1000 km^2", axis(3) placement(nw))
   clpattern(dash)
   || line cil winter, yaxis(2) yscale(range(-4,3) axis(2))
   ylabel(, nogrid axis(2))
   ytitle("CIL temperature, degrees C", axis(2)) clpattern(solid)
   || connected cod winter, yaxis(1) yscale(range(0,200) axis(1))
   ylabel(, nogrid axis(1))
   ytitle("Cod catch, 1000 tons", axis(1) placement(sw))
   || if winter > 1959,
   legend(ring(0) position(7) label(1 "Max ice area")
      label(2 "Min CIL temp") label(3 "Cod catch") rows(3))
   xtitle("") xlabel(1960(5)2000, grid)
   saving(c:\data\fig03_48.gph, replace) ;
graph export c:\data\fig03_48.eps, replace ;
#delimit cr
```

这个 do 文件的第一行设定用英文分号(;)作为一行结束的分隔符。因此,Stata 此后直到遇到一个分号时,才会认为一行已经结束。第二行只是重新读入画出图 3.48 所需的数据集(*gulf.dta*);请注意结束本行的就是分号。较长的 graph twoway 命令占用了接下来的 15 行,但是 Stata 将这些行看作是以选项 saving( )后的分号表示结束的一个逻辑行。该选项将图形存成 Stata 的.gph 格式文件。

接下来,命令 **graph export** 以封装后记格式(Encapsulated Postscript format)创建了同一图形的另一个版本,正如文件名 *fig04_84.eps* 中的扩展名.eps 所表明的那样。(键入 **help graph_export** 了解该命令的更多内容,该命令对于编写将重复创建图形的程序或 do 文件特别有用)

do 文件最后的 #delimit cr 命令重新设定回车作为行结束的分隔符,以回到 Stata 的常规模式。尽管它在纸上是不可见的,#delimit cr 这一行自己必须以回车(提示:即 Enter 键)来结束,于是在 do 文件的结尾创建了最后的一个空白行。

取出与合并图形

通过 **graph use** 命令,任何保存为 Stata 中"活的"(live) .ghp 格式的图形随后都能取到内存中来。比如,要取出图 3.48,我们键入以下命令:

```
. graph use fig03_48
```

一旦图形在内存中,它就被显示在屏幕上,并能被打印或者再次以不同的名称或格式加以保存。从先前以 .gph 格式保存的图形,我们随后可以存成诸如后记(Postscript,扩展名为 .ps)、便携网络图形(Portable Network Graphics,扩展名为 .png)以及增强型视窗图元文件(Enhanced Windows metafile,扩展名为 .emf)等其他格式的版本。我们也可以通过菜单、或者直接在 **graph use** 命令中改变颜色方案(color scheme)。图形 *fig03_48.gph* 是采用 s2 单色方案加以保存的,但是我们可以通过键入以下命令来看看它用 s1 颜色方案会怎么样。

```
. graph use fig03_47, scheme(s1color)
```

保存在磁盘上的图形也可以通过 **graph combine** 命令加以合并。这提供了一种将多个图形合并成同一图像的方式。为了举例说明这点,我们回到前面在图 3.48 中显示的圣劳伦斯湾数据。以下命令画出了三幅简单的时间图(未显示),将它们以 *fig03_49 a.gph*、*fig03 _49 b.gph* 和 *fig03 _49 c.gph* 的名称加以保存。子选项 **margin(medium)** 设定每幅图中标题文本框的边距宽度。

为了合并这几幅图形,我们键入以下命令。因为三幅图具有相同的 *x* 尺度,因此将这些图形垂直排列成三行是有意义的。选项 **imargin** 设定图 3.49 中每幅图周围的"极小"边距。

键入 **help graph_combine** 查看有关这一命令的更多信息。选项控制还包括行数或列数、文本和标志的大小(否则会随着图形数量的增多而变得太小)以及各个图形之间的边距等细节。它们也可以设定二维图的 *x* 或 *y* 轴是否具有共同的尺度,或者对所有的成分指定一个共同的颜色方案。标题也能被加入到一幅合并的图形中,该图能被打印、保存、取回或者为了再次以通常的方式进行合并。

```
. graph twoway line maxarea winter if winter > 1964, xtitle("")
      xlabel(1965(5)2000, grid) ylabel(50(50)200, nogrid)
      title("Maximum winter ice area", position(4) ring(0) box
         margin(medium))
      ytitle("1000 km^2") saving(fig03_49a)

. graph twoway line cil winter if winter > 1964, xtitle("")
      xlabel(1965(5)2000, grid) ylabel(-1(.5)1.5, nogrid)
      title("Minimum CIL temperature", position(1) ring(0) box
         margin(medium))
      ytitle("Degrees C") saving(fig03_49b)

. graph twoway line cod winter if winter > 1964,  xtitle("")
      xlabel(1965(5)2000, grid) ylabel(0(20)100, nogrid)
      title("Northern Gulf cod catch", position(1) ring(0) box
         margin(medium))
      ytitle("1000 tons") saving(fig03_49c)

. graph combine fig03_49a.gph fig03_49b.gph fig03_49c.gph,
      imargin(vsmall) rows(3)
```

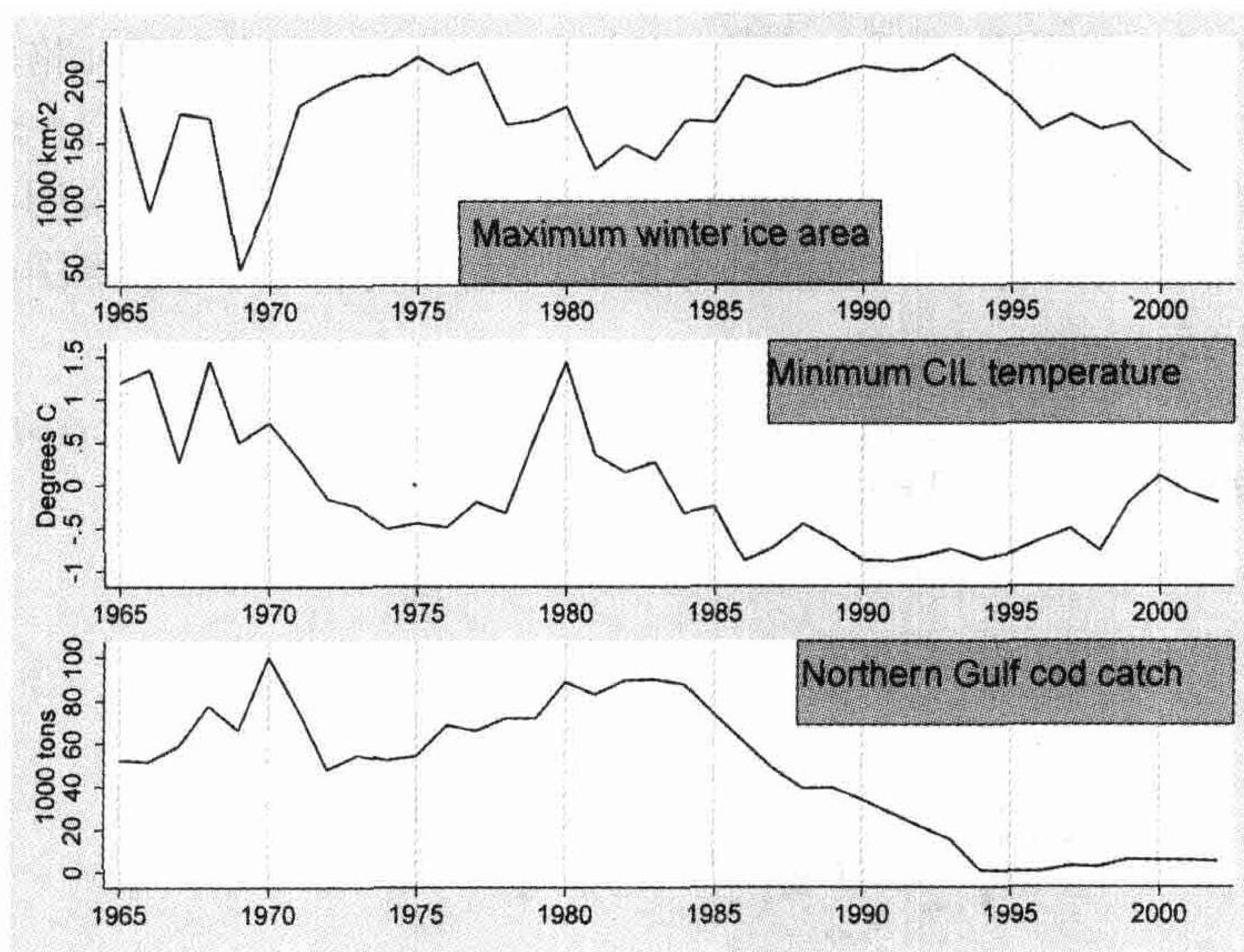


图 3.49

我们最后的例子示范了好几个这种 **graph combine** 选项，并且建立具有不等规模分图的合并图。假如我们想要一幅类似于前面图 3.42 那样的吸烟与大学毕业生的散点图，但是还要在各自坐标轴旁画出 *y* 和 *x* 变量的箱线图。使用 *statehealth.dta*，我们可以尝试着使用以下命令首先画出一幅 *smokeT* 的垂直箱线图、一幅 *smokeT* 对 *college* 的散点图以及一幅 *college* 的水平箱线图，最后再将这三幅图合并成一个图像（未显示）。

```
. graph box smokeT, saving(wrong1)

. graph twoway scatter smokeT college, saving(wrong2)

. graph hbox college, saving(wrong2)

. graph combine wrong1.gph wrong2.gph wrong3.gph
```

但是，由以上命令得到的合并图形看起来是错误的。我们将得到两幅宽大的箱线图，每一幅都与散点图一样大，并且其坐标轴也未加以对齐。为了得到一个更为满意的

版本,我们需要从创建一幅 *smokeT* 的很窄的垂直箱线图开始。以下命令中的选项 **fxsize(20)** 将图形的 *x*(横轴)固定为正常幅度的20%,这将导致一个正常高度但只有20% 宽度的图形。考虑到在最终图形中将会出现的间隔,还增加了两行空白的说明。

```
. graph box smokeT, fxsize(20) caption(""   "")
      ytitle("") ylabel(none) ytick(15(5)35, grid) saving(fig03_50a)
```

对第二个成分,我们创建一幅 *smokeT* 对 *college* 的简单散点图。

```
.  graph twoway scatter smokeT college,
      ytitle("Percent adults smoking")
      xtitle("Percent adults with Bachelor's degrees or higher")
      ylabel(, grid) xlabel(, grid) saving(fig03_50b)
```

第三个成分是一幅 *college* 的很扁的水平箱线图。该图应当具有正常的宽度,但是 *y*(纵轴)固定在正幅度的20% 。为了保持间隔,也增加了两列空白的左标题

```
.  graph hbox college, fysize(20) l1title("") l2title("")
      ylabel(none) ytick(10(5)35, grid) ytitle("") saving(fig03_50c)
```

这三个成分在图3.50 中合并在一起。**graph combine** 命令的选项 **cols(2)** 将图形排成两列,就像有一个空单元格的2 乘2 表格。选项 **holes(3)** 设定空单元格应当是第三个,因此我们的三个成分图形填入了位置1,2 和4。出于可读性的考虑,**iscale(1.05)** 将标志记号和文本放大了大约5% 。我们在初始箱线图中增加的空白说明和标题弥补了散点图每一坐标轴上的两行文本(标题和标签),因此两个箱线图与散点图的坐标轴对齐了(尽管不是很完美)。

```
.  graph combine fig03_50a.gph fig03_50b.gph fig03_50c.gph,
      cols(2) holes(3) iscale(1.05)
```

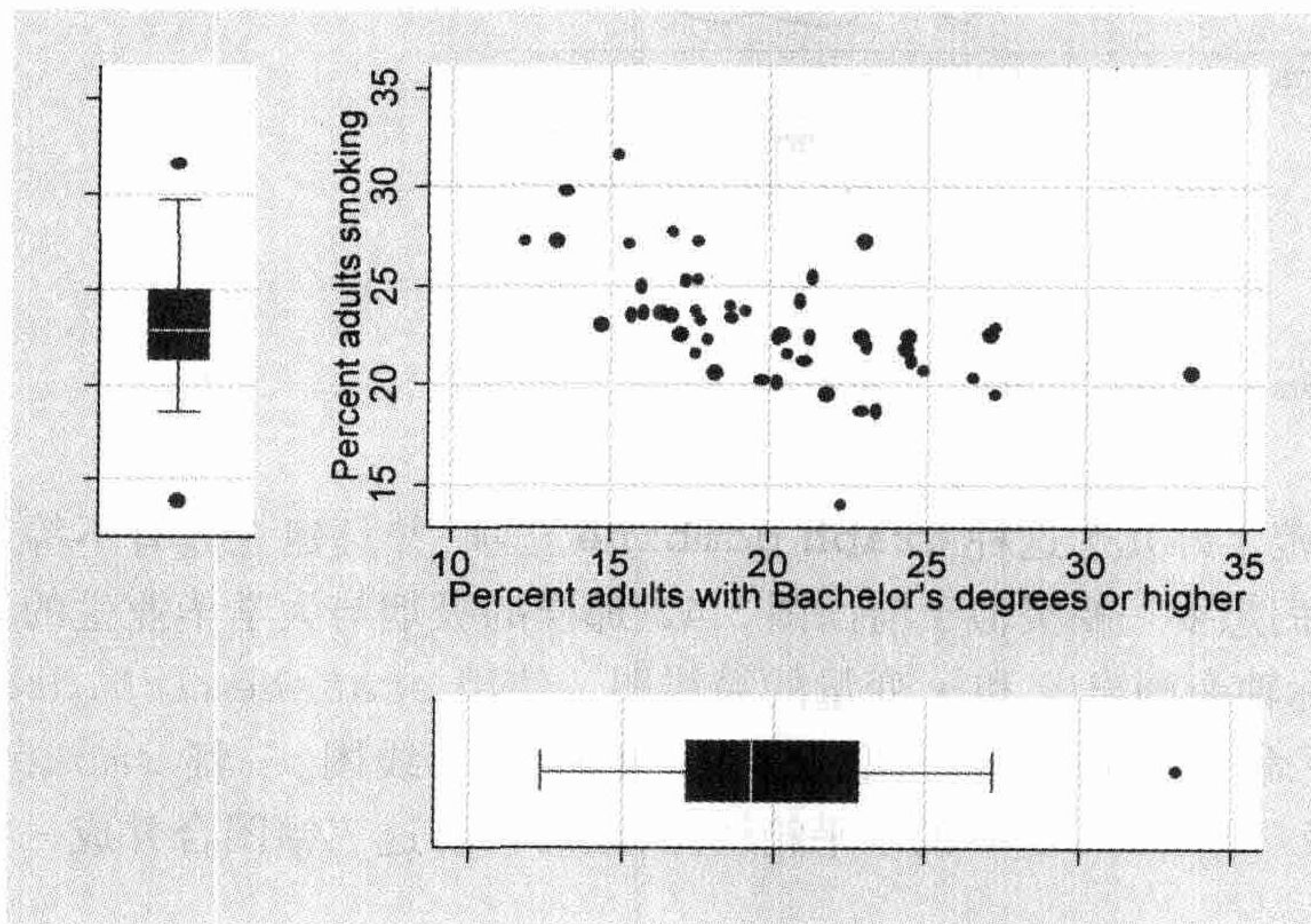


图3.50

4 概要统计及交互表

命令 **summarize** 可以对测量型变量做简单的描述性统计，比如，计算变量的中位数、平均数及标准差。而 **tabstat** 命令则可以进行更加灵活的概要统计。对于定类或者定序变量，**tabulate** 命令可以获得频数分布表、交互表、分类检验以及进行关联度测量，此外，**tabulate** 也可以根据其他变量的类别创建有关平均值及标准差的一维或者二维表格。**table** 作为一般性的表格创建命令，可以创建多达七维的交互表，表中的单元格包含诸如频数、总和、平均数或者中位数等统计量。在本章的最后，我们将进一步探讨单变量的操作程序，包括正态性检验、变量转换以及展示探索性数据分析(EDA)。本章所涉及的大部分分析既可以通过所列出来的命令来完成，也可以通过菜单进行选择：**Statistics-Summaries, tables & tests**。

Stata 除了这些一般目的的分析方法之外，还提供许多流行病学家感兴趣的交互表方法。虽然本章没有对此进行描述，但是读者可以通过键入 **help epitab** 查看相关的信息。Selvin (1996)对此也进行过介绍。

命令示范

```
. summarize y1 y2 y3
```

对所列变量计算简单的概要统计指标(平均数、标准差、最小值和最大值、观测案例数)。

```
. summarize y1 y2 y3, detail
```

获取详细的描述性统计，包括百分位数、中位数、平均数、标准差、方差、偏度、峰度等。

```
. summarize y1 if x1 > 3 & x2 < .
```

只计算 *x1* 大于 3 且 *x2* 不是缺失值条件下的变量 *y1* 的概要统计指标。

```
. summarize y1 [fweight = w], detail
```

利用变量 *w* 作为加权变量进行频数加权，计算 *y1* 详细的概要统计。

```
. tabstat y1, stats(mean sd skewness kurtosis n)
```

只计算变量 *y1* 的具体指定的概要统计指标。

```
. tabstat y1, stats(min p5 p25 p50 p75 p95 max) by(x1)
```

按变量 *x1* 的每个类别,分别计算测量型变量 *y1* 的具体指定的概要统计(最小值、第 5 百分位数、第 25 百分位数,等等)。

```
. tabulate x1
```

显示变量 *x1* 所有非缺失值的频数分布表。

```
. tabulate x1, sort miss
```

显示 *x1* 所有值的频数分布,包括缺失值。同时,按频数大小对行(变量值)进行排序。

```
. tab1 x1 x2 x3 x4
```

对所列变量分别创建频数分布表。

```
. tabulate x1 x2
```

显示一个两变量交互表,其中 *x1* 为行变量,*x2* 为列变量。

```
. tabulate x1 x2, chi2 nof column
```

创建一个交互表,并对两变量的独立性进行皮尔逊卡方(Pearson χ^2) 检验。每一单元格内不再显示频数而是给出列百分比。

```
. tabulate x1 x2, missing row all
```

创建一个交互表,在计算频数和百分比时把缺失值包括在内。同时,计算"所有"可用的统计量(皮尔逊卡方(Pearson χ^2) 和似然比卡方(likelihood-ratio χ^2), Cramer的 V 检验, Goodman 和 Kruskal 的 gamma 检验以及 Kendall 的 τ_b 检验)

```
. tab2 x1 x2 x3 x4
```

创建所列变量的所有可能的二维交互表。

```
. tabulate x1, summ(y)
```

创建一个一维表,显示 *x1* 每个类别中变量 *y* 的均值、标准差及频数。

```
. tabulate x1 x2, summ(y) means
```

创建一个二维表,显示 *x1* 和 *x2* 每一种组合下 *y* 的均值。

```
. by x3, sort:  tabulate x1 x2, exact
```

创建一个三维交互表,在 *x3* 的每一个取值下创建 *x1*(行)和 *x2*(列)的分表,并为每个分表计算费舍精确检验(Fisher's exact test)。命令 **by varname, sort:**几乎可以作为任何有意义的 Stata 命令的前缀发挥作用。如果数据已经根据变量名 *varname*进行了排序,那么这里的 **sort** 选项就没有必要了。

```
. table y x2 x3, by(x4 x5) contents(freq)
```

创建一个五维交互表,其中 *x4* 为大行 1、*x5* 为大行 2,*x3* 为大列,构成基础三维表,然后其中每个交互单元都是 *y*(行)和 *x2* (列)的二维交互表,每个单元格包含频数。

```
. table x1 x2, contents(mean y1 median y2)
```

创建 *x1* (行)和 *x2* (列)的二维交互表,单元格包含 *y1* 的平均数和 *y2* 的中位数。

定距变量的描述性统计

数据集 *VTtown.dta* 包含了美国佛蒙特州某镇居民的信息。例行的州检验发现,在

供水中有微量的有毒化学物质,随后对此专门做了一项调查。在几口私井里和公立学校附近,有毒化学物质的浓度最高。担忧的市民召开了好几次会议讨论这一问题的可能解决途径。

```
Contains data from C:\data\VTtown.dta
  obs:           153                         VT town survey (Hamilton 1985)
 vars:             7                         11 Jul 2005 18:05
 size:         1 683 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
gender          byte   %8.0g       sexlbl     Respondent's gender
lived           byte   %8.0g                  Years lived in town
kids            byte   %8.0g       kidlbl     Have children <19 in town?
educ            byte   %8.0g                  Highest year school completed
meetings        byte   %8.0g       kidlbl     Attended meetings on pollution
contam          byte   %8.0g       contamlb   Believe own property/water
                                                contaminated
school          byte   %8.0g       close      School closing opinion
-------------------------------------------------------------------------------
Sorted by:
```

要得到变量 *lived*(受访者在本镇所住年数)的平均值和标准差,键入:

```
. summarize lived

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       lived |       153    19.26797    16.95466          1         81
```

这个表还给出了非缺失观测值的数目以及变量的最小值和最大值。如果我们只键入 **summarize** 而没有列出变量,我们将获得数据集中每一个数值变量的平均数和标准差。

要想看到更详细的概要统计,键入

```
. summarize lived, detail

                        Years lived in town
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            2              1
10%            3              1       Obs                 153
25%            5              1       Sum of Wgt.         153

50%           15                      Mean           19.26797
                        Largest       Std. Dev.      16.95466
75%           29             65
90%           42             65       Variance       287.4606
95%           55             68       Skewness       1.208804
99%           68             81       Kurtosis       4.025642
```

这个 **summarize, detail** 命令输出包括基本统计量再加上以下各项指标:

百分位数(percentiles): 特别是第一个四分位数(quartile,即第 25 百分位数)、中位数(median,即第 50 百分位数)和第三个四分位数(第 75 百分位数)。因为许多样本数据不能平等地进行四分或者其他的标准划分,这些百分位数通常是近似值。

四个最小值和四个最大值: 通常可以显示出特异值。

权数和(sum of weights): Stata 能识别四种类型的权数:分析权数(**aweight**)、频数权数(**fweight**)、重要性权数(**iweight**)、抽样权数(**pweight**)。不同的程序认可和允许不同的权数。比如, **summarize, detail** 命令允许使用

	aweight 或者 **fweight** 。用 **help weights** 命令可以看到更多这方面的说明。
方差(variance):	标准差(standard deviation)的平方(更恰当地说,标准差等于方差的平方根)。
偏度(skewness):	不对称的方向和程度。一个十分对称分布的偏度值等于0。正偏度(右边尾巴拖得较长)的偏度值大于0;负偏度(左边尾巴拖得较长)的偏度值小于0。
峰度(kurtosis):	尾重(tail weight)。正态(normal,或称高斯,Gaussian)分布是对称分布,其峰度值等于3。如果一个对称分布有着比正态分布更长的尾巴(即呈尖峰状),那么它的峰度值 >3。如果峰度值 <3 则表明比正态分布的尾巴短。

tabstat 命令也可以做概要统计,但是它比 **summarize** 更灵活。我们可以明确规定我们想获取的概要统计指标。例如:

```
. tabstat lived, stats(mean range skewness)

    variable |       mean     range  skewness
-------------+------------------------------
       lived |   19.26797        80  1.208804
--------------------------------------------
```

加一个 **by**(**varname**) 选项,**tabstat** 便会创建一个表格,它包括 *varname* 每个取值下的概要统计。下面这个例子包括 *gender* 每一类下变量 *lived* 的平均数、标准差、中位数、四分位距(interquartile range)以及非缺失观测值的数目。平均数和中位数都表明样本中的妇女居住本镇的平均时间比男性短好几年。需要注意的是中位数一栏被标为“p50”,意思是第50百分位数。

```
. tabstat lived, stats(mean sd median iqr n) by(gender)

Summary for variables: lived
     by categories of: gender (Respondent's gender)

gender |      mean        sd       p50       iqr         N
-------+--------------------------------------------------
  male |  23.48333  19.69125      19.5        28        60
female |  16.54839  14.39468        13        19        93
-------+--------------------------------------------------
 Total |  19.26797  16.95466        15        24       153
----------------------------------------------------------
```

tabstat 命令的 **stats**( )选项可以包括下列统计量:

mean	平均数
count	非缺失观测值总数
n	等同于计数 **count**
sum	总和
max	最大值
min	最小值
range	极差 = 最大值-最小值
sd	标准差
var	方差
cv	变异系数 = 标准差 /平均值
semean	标准误 = 标准差 $/\sqrt{n}$
skewness	偏度

kurtosis	峰度
median	中位数(等同于第50百分位数,**p50**)
p1	第1百分位数(同理,**p5**、**p10**、**p25**、**p50**、**p75**、**p95** 或 **p99**)
iqr	四分位距 = 第75百分位数 - 第25百分位数
q	四分位数;等于指定了第25、第50和第75分位数

另外,**tabstat** 的选项还可以控制表格输出的格式和标签。要想查看 **tabstat** 选项的完整列表,请键入 **help tabstat** 。

summarize 或者 **tabstat** 命令取得的统计指标对当前的样本进行了描述。我们也可能想对总体进行推断,比如,创建变量 *lived* 平均值的99% 置信区间:

```
. ci lived, level(99)

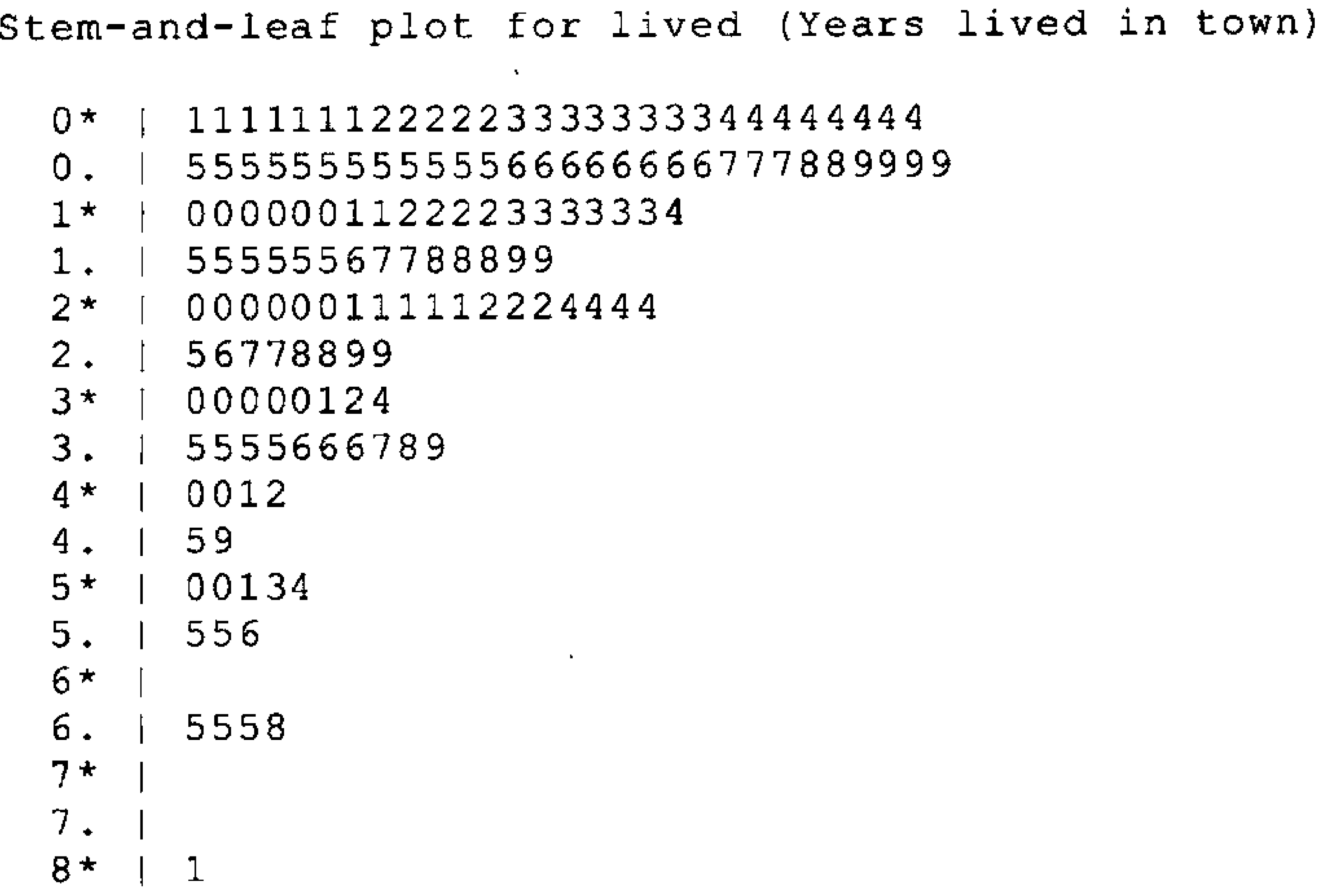
    Variable |        Obs        Mean    Std. Err.       [99% Conf. Interval]
-------------+---------------------------------------------------------------
       lived |        153    19.26797    1.370703        15.69241    22.84354
```

基于这个样本,我们有99% 的信心认为总体的平均值会落在15.69 ~22.84 年这个区间内。在这里,我们用 **level( )** 选项具体指定一个99% 的置信区间。如果我们忽略这个选项,**ci** 命令会默认做出一个95% 的置信区间。

使用其他选项可以使 **ci** 计算那些服从二项分布或者泊松分布的变量的精确置信区间。与 **ci** 相关的一个命令是 **cii** ,它可以直接根据概要统计指标,比如,我们可能会在发表的论文中碰到的那些,来计算正态分布、二项分布或者泊松分布的置信区间,而它并不需要原始数据。可以键入 **help ci** 获取有关这两个命令的详细内容。

探测性数据分析

统计学家 John Tukey 发明了一整套探测性数据分析(exploratory data analysis,简称EDA)方法,不用一些无关紧要的假定,用一种探测性地和怀疑地方式来分析数据(见 Tukey,1977;以及 Hoaglin, Mosteller 和 Tukey,1983, 1985)。我们在第3章介绍的箱线图就是 Tukey 最有名的发明之一。Tukey 的另一项发明则是茎叶图,这是一种对数据值进行排序的图,每一观测数据值的前位数构成了"茎",而后位数则组成了"叶"。



```
. stem lived

Stem-and-leaf plot for lived (Years lived in town)

  0* | 1111111222223333333344444444
  0. | 55555555555566666666777889999
  1* | 0000001122223333334
  1. | 55555567788899
  2* | 000000111112224444
  2. | 56778899
  3* | 00000124
  3. | 5555666789
  4* | 0012
  4. | 59
  5* | 00134
  5. | 556
  6* |
  6. | 5558
  7* |
  7. |
  8* | 1
```

在这里,**stem** 自动选择了一个双茎形式的茎叶图,其中 1* 表示首位数是 1 和第二位数是 0 ~4(也就是指那些已经在本镇住了 10 ~14 年的受访者)。1. 则表示首位数是 1 和第二位数是 5 ~9(15 ~19 年)。我们可以用 **lines( )** 选项控制每个首位数的列的数量。例如,在一个五茎形式的茎叶图中,1* 表示持有叶子 0 ~1,1t 表示持有叶子 2 ~3,1f 表示持有叶子 4 ~5,1s 表示持有叶子 6 ~7,1. 表示持有叶子 8 ~9。要获取这种五茎形式的茎叶图,键入:

```
. stem lived, lines(5)
```

键入 **help stem** 可以得到有关其他选项的信息。

字符数值表(**lv**)利用序次统计量来解剖分布状况。

```
. lv lived

#      153              Years lived in town
              ----------------------------------
M      77    |                 15              |      spread   pseudosigma
F      39    |      5          17          29  |        24       17.9731
E      20    |      3          21          39  |        36       15.86391
D      10.5  |      2          27          52  |        50       16.62351
C       5.5  |      1       30.75        60.5  |      59.5       16.26523
B       3    |      1          33          65  |        64       15.15955
A       2    |      1        34.5          68  |        67       14.59762
Z       1.5  |      1       37.75        74.5  |      73.5       15.14113
        1    |      1          41          81  |        80       15.32737
             |                                 |
             |                                 |   # below       # above
inner fence  |    -31                      65  |         0             5
outer fence  |    -67                     101  |         0             0
```

M 表示中位数,**F** 表示"四分位数"(与 **summarize, detail** 和 **tab,sum** 不同,它采用另一种近似计算方法来计算四分位数)。**E**、**D**、**C** 等分别表示大约在 1/8、1/16、1/32 等分割点之外的分布情况。第二列数给出了每个字符值的"深度"或者说是离最近极值的距离。在图中心的方框内,中间的一列给出了"中间概要值",它们分别是两个字符值的平均值。如果中间概要值背离中位数,就像上图变量 *lived* 显示的那样,就告诉我们此变量分布越接近尾部的部分就越偏。"spread"(展开)这一列表示每对字符值的差,比如,F 的"spread"等于近似的四分位距。最后,右端一列的"pseudosigma"(伪 σ)计算的是假定这些字符值描述的是假定为正态分布总体时的标准差。F 行的伪 σ 有时又被称作"伪标准差"(pseudo standard deviation,简标为 *PSD*),它是针对对称分布的近似正态性问题提供的一种简易的而且抗特异值影响的检验。

①比较平均数和中位数以诊断总体的偏态:

平均数 > 中位数:正偏态

平均数 = 中位数:对称

平均数 < 中位数:负偏态

②假如平均数和中位数近似,则表明对称,然后比较标准差和伪标准差 *PSD*,这有助于我们评估尾部正态性:

标准差 > *PSD*:比正态分布的尾重大(heavier-than-normal tail)

标准差 = *PSD*:正态分布尾重(normal tail)

标准差 < *PSD*:比正态分布的尾重轻(lighter-than-normal tail)

令 F_1 和 F_3 分别表示第 1 个和第 3 个四分位数(近似于第 25 百分位数和第 75 百分位数)。

那么,四分位距 *IQR* 等于 $F_3 - F_1$,并且有 $PSD = IQR/1.349$。

命令 **lv** 也可以识别那些轻度和严重的特异值。当一个 x 的值处在内栅栏之外但还在外栅栏之内时,我们称之为"轻度特异值":

$$F_1 - 3IQR \leqslant x < F_1 - 1.5IQR \quad \text{或者} \quad F_3 + 1.5IQR < x \leqslant F_3 + 3IQR$$

假如 x 的值处在外栅栏之外,那么它就是一个"严重的特异值":

$$x < F_1 - 3IQR \quad \text{或者} \quad x > F_3 + 3IQR$$

lv 命令给出了这些分割点以及每种类型的特异值的数目。外栅栏之外的严重的特异值在正态总体中很少发生(大约为百万分之二)。蒙特卡罗模拟显示出,从规模为 15 至规模约为 20 000 的样本中,任何严重特异值的存在都足以成为在显著水平为 0.05 的条件下拒绝正态性假设的充分证据(Hamilton, 1992b)。严重的特异值会给许多的统计技术带来问题。

summarize、**stem** 和 **lv** 都证实变量 *lived* 的样本分布为正偏态,根本不像理论上的正态曲线。下一节将介绍更为正规的正态性检验以及能减少变量偏态的数据转换方法。

正态性检验和数据转换

许多的统计程序只有在变量服从正态分布时才能工作得最好。上一节我们所介绍的检验近似正态性的探索性方法扩展了在第 3 章所介绍的图形工具(直方图、箱线图以及分位—正态图)。有一种更为正规的偏度—峰度检验,它利用命令 **summarize, detail** 显示的偏度和峰度统计值来检验虚无假设,即手头的样本是来自一个正态分布总体。

```
. sktest lived

                   Skewness/Kurtosis tests for Normality
                                                 ------- joint ------
    Variable |  Pr(Skewness)   Pr(Kurtosis)  adj chi2(2)     Prob>chi2
-------------+-------------------------------------------------------
       lived |       0.000          0.028          24.79        0.0000
```

这里,**sktest** 拒绝了正态性假设:变量 *lived* 无论是在偏度(P = 0.000)还是在峰度(P = 0.028),抑或是把两者结合在一起考虑(P = 0.000 0)都表现出显著的非正态。Stata 只显示保留三位或者四位小数的概率,"0.000 0" 其实表明 P < 0.000 05。

其他的正态性检验或者对数正态性检验包括 Shapiro-Wilk 的 W 检验(**swilk**)以及 Shapiro-Francia 的 W' 检验(**sfrancia**)。键入 **help sktest** 查看这些选项的信息。

数据的非线性转换,比如,取平方根和求对数,经常被用于改变分布的形态,其主要目标是使那些偏态分布更加对称,也许就更接近于正态。转换也有助于确立变量之间的线性关系(第 8 章)。表 4.1 展示了一种被称作"幂阶梯"(ladder of powers)(Tukey,1977)的升级方法,它可以指导我们选择合适的转换以改变分布的形态。变量 *lived* 呈现出轻度的正偏态,因此它的平方根可能会更加对称。我们可以通过键入下列命令创建一个新变量,使之等于变量 *lived* 的平方根:

```
. generate srlived = lived ^.5
```

我们也可以用 **sqrt(*lived*)** 来代替 ***lived*^.5**,两者作用完全相同。

取对数是另外一种转换,它可以降低正偏态。要想创建一个新变量,使它等于变量

lived 的自然对数(底为 *e*),键入:

```
. generate loglived = ln(lived)
```

在幂阶梯方法及相关的转换方法(如 Box-Cox 方法)中,对数其实是取代了“0”次方的位置,它们对于分布形态的影响介于 0.5 次方(即平方根)和 -0.5 次方(即平方根倒数)转换之间。

表 4.1 幂阶梯

Stata 输出标签	转 换	公 式	效 果
cube	立方	*new* = *old*^3	减少严重负偏态
square	平方	*new* = *old*^2	减少轻度负偏态
raw	原始	*old*	没有变化(原始数据)
square-root	平方根	*new* = *old*^.5	减少轻度正偏态
log e (or log 10)	自然对数 $\log_e$ (或 $\log_{10}$)	new = ln(*old*) *new* = *log*10(*old*)	减少正偏态
negatine reciprocal root	平方根负倒数	*new* = -(*old*^-.5)	减少严重正偏态
negatine reciprocal	负倒数	*new* = -(*old*^-1)	减少非常严重正偏态
negatine reciprocal square	平方负倒数	*new* = -(*old*^-2)	同上
negatine reciprocal cube	立方负倒数	*new* = -(*old*^-3)	同上

当我们在乘方次数从负数向 0 提高时,那么取结果的负值便保留了原来的顺序,即原变量 *old* 的值越大,其转变后新变量 *new* 的值也就越大,反之亦然。当原变量 *old* 本身包含负值或零时,在进行转换之前必须先加一个常数。例如,假如变量 *arrests* 测量的是一个人曾经被逮捕的次数(很多人为 0 次),那么一种比较合适的对数转换是:

```
. generate larrests = ln(arrests + 1)
```

ladder 命令把幂阶梯和 **sktest** 的正态性检验结合在一起。它尝试幂阶梯上的每一种幂并报告其结果是否显著性地非正态。我们可以利用具有严重偏态的变量 *energy*(人均能源消耗)来加以展示,*energy* 这个变量来自数据集 *states.dta*。

```
. ladder energy

Transformation         formula              chi2(2)       P(chi2)
------------------------------------------------------------------
cube                   energy^3              53.74         0.000
square                 energy^2              45.53         0.000
raw                    energy                33.25         0.000
square-root            sqrt(energy)          25.03         0.000
log                    log(energy)           15.88         0.000
reciprocal root        1/sqrt(energy)         7.36         0.025
reciprocal             1/energy               1.32         0.517
reciprocal square      1/(energy^2)           4.13         0.127
reciprocal cube        1/(energy^3)          11.56         0.003
```

从结果看,倒数转换 1/*energy*(即 $energy^{-1}$)最接近于正态分布。其他的大部分转换(包括原始数据)都显著性地非正态。图 4.1(由命令 **gladder** 创建)通过比较每一种转换的直方图与正态曲线,直观地支持了这个结论。

```
. gladder energy
```

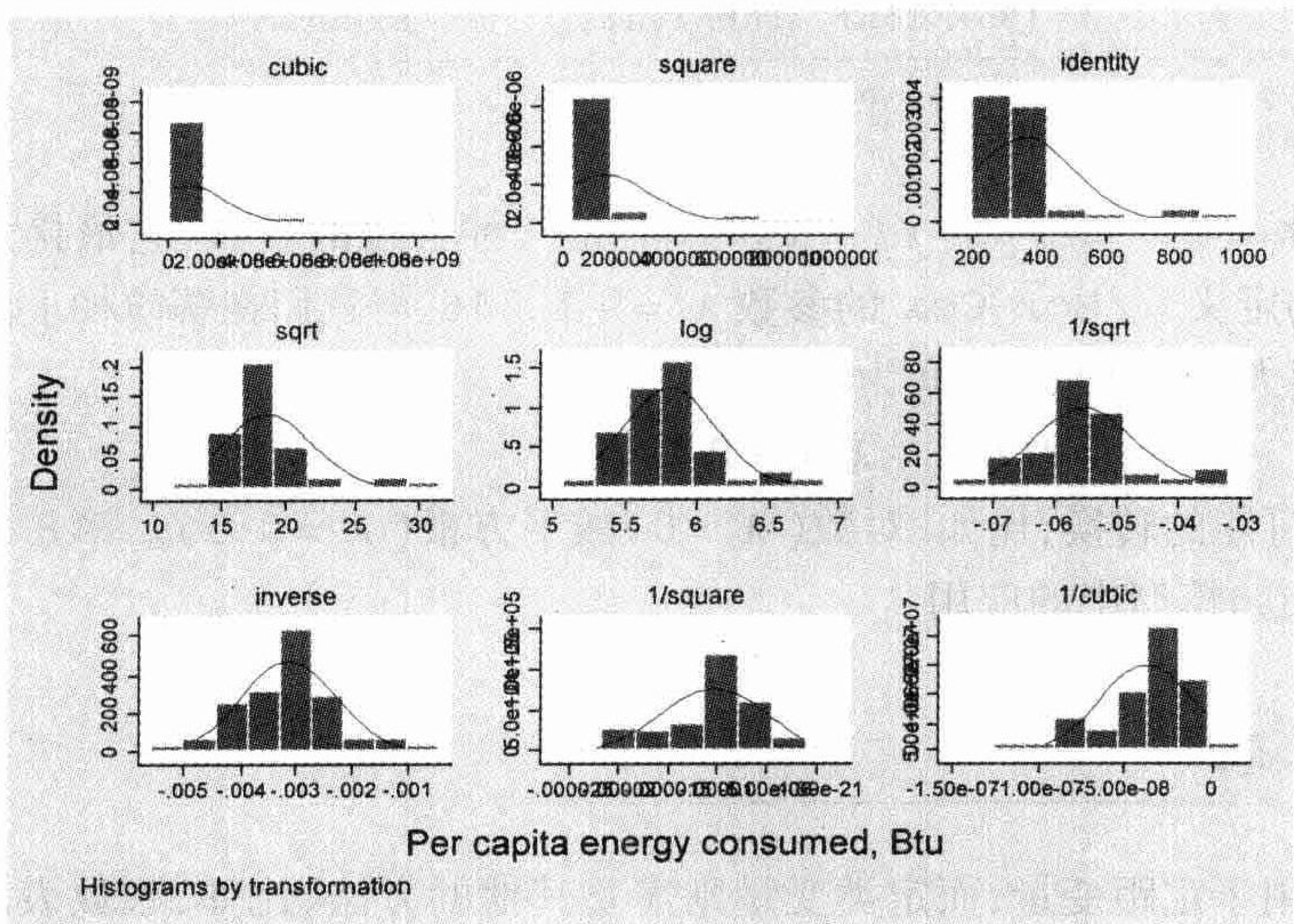


图 4.1

图 4.2 显示的是一组相应的幂阶梯转换的分位—正态图，它们是通过"四分位阶梯"的命令 **qladder** 取得的。在这个例子中，为了使这些小图更具可读性，我们可以使用选项 **scale(1.25)** 按 25% 比例增大绘制标签和符号。坐标轴标签可以通过选项 **ylabel(none) xlabel(none)** 进行隐藏。

```
. qladder energy, scale(1.25) ylabel(none) xlabel(none)
```

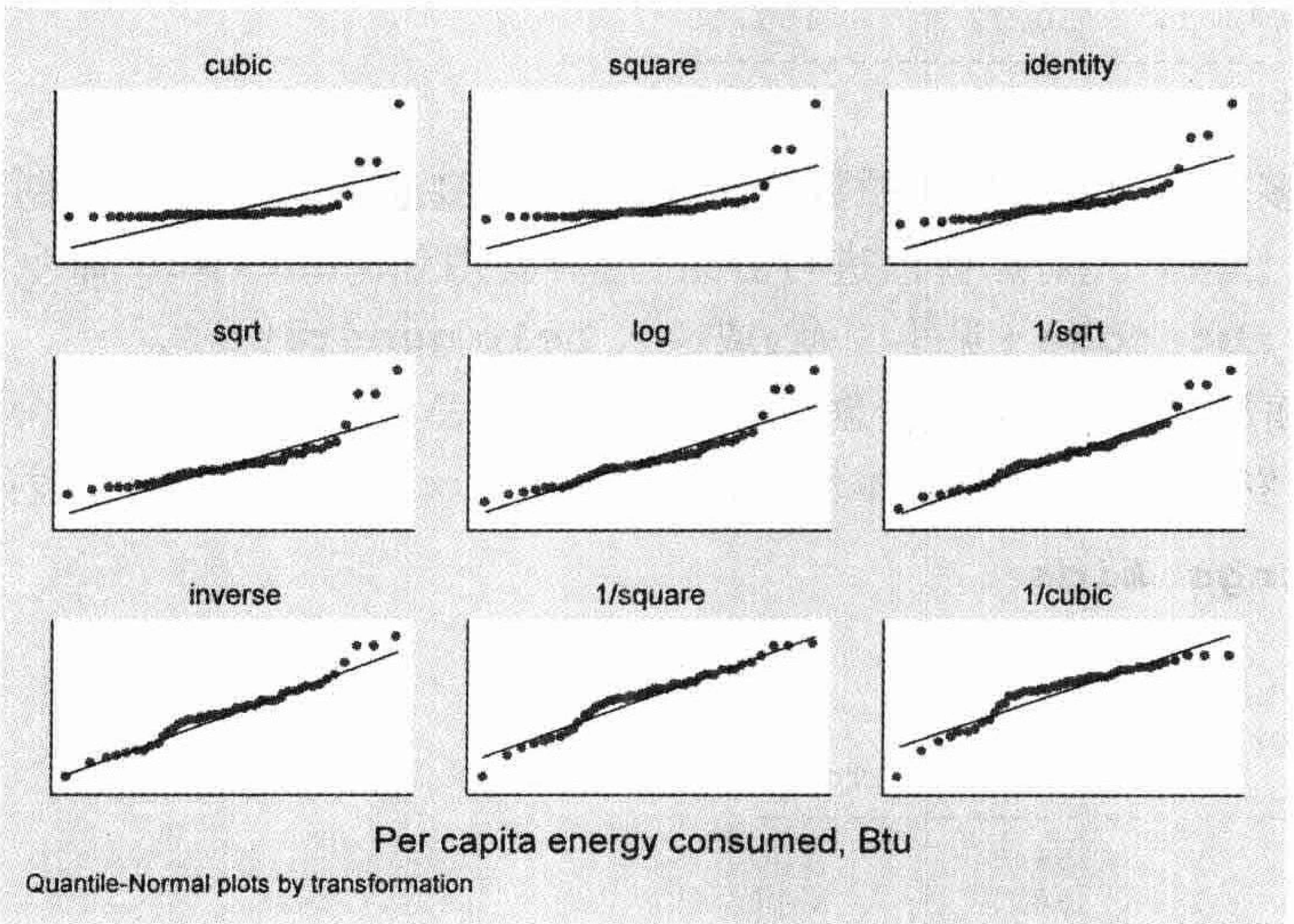


图 4.2

另一项转换技术就是 Box-Cox 转换，它可以自动在转换方法中选择（对于分析者来说比较容易，但并不总是一件好事）并提供细微的层次。命令 **bcskew0** 可以为Box-Cox 转换寻找 λ 值，

$$y^{(\lambda)} = \{y^{\lambda} - 1\}/\lambda, \qquad \text{当}\lambda > 0 \text{或} \lambda < 0$$

或者

$$y^{(\lambda)} = \ln(y), \qquad \text{当}\lambda = 0$$

这样便使 $y^{(\lambda)}$ 的偏度近似于 0。把它应用到变量 *energy*，我们得到转换变量 *benergy*：

```
. bcskew0 benergy = energy, level(95)
```

```
       Transform |          L      [95% Conf. Interval]       Skewness
-----------------+--------------------------------------------------------
  (energy^L-1)/L |  -1.246052    -2.052503  -.6163383         .000281
(1 missing value generated)
```

就是说，*benergy* =(*energy*$^{-1.246}$ -1)/(-1.246) 是一种合适的转换，它最接近于对称（根据偏度统计量的定义）。Box-Cox 的参数 λ = -1.246 与我们在幂阶梯上的选择（-1 阶）的差别不是很大。λ 的置信区间为：

$$-2.0525 < \lambda < -0.6163$$

它使我们拒绝其他一些可能的转换，比如，对数（λ =0）或平方根（λ =0.5）。第 8 章将介绍 Box-Cox 方法在回归模型中的应用。

频数表和二维交互表

上面介绍的方法适用于定距变量，而定类变量则需要其他的方法，比如，频数表。返回到前面所用的数据集 *VTtown.dta*，通过定类变量 *meetings* 频数表，我们可以取得参加有关污染问题会议的受访者比例。

```
. tabulate meetings

   Attended |
meetings on |
  pollution |      Freq.     Percent        Cum.
------------+-----------------------------------
         no |        106       69.28       69.28
        yes |         47       30.72      100.00
------------+-----------------------------------
      Total |        153      100.00
```

tabulate 可以为有极多取值的变量创建频数分布表。不过，要想为这样的变量创建一个易操控的频数分布表，你最好先对那些值进行分组，这需要用到 **generate** 命令以及它的选项 **recode** 或者 **autocode**（见第 2 章，或键入 **help generate**）。

tabulate 命令后面加两个变量名就会创建一个二维交互表。例如，这里有一个根据 *kids*（受访者是否有 19 岁以下的孩子住在本城）所创建的关于 *meetings* 的交互表：

```
.  tabulate meetings kids

  Attended |
  meetings | Have children <19 in
        on |        town?
 pollution |        no        yes |     Total
-----------+----------------------+----------
        no |        52         54 |       106
       yes |        11         36 |        47
-----------+----------------------+----------
     Total |        63         90 |       153
```

在上表中，第一个列出的变量构成了表的行，而第二个变量构成了表的列。我们可以看到在这 153 名受访者中只有 11 人既没有 19 岁以下的孩子住在本城，同时又参加了会议。

tabulate 有许多对创建频数表非常有用的选项：

all　　等同于选项中包括 **chi2**、**lrchi2**、**gamma**、**taub**、**V** 各项。对于一个特定的频数表来说，并不是所有的这些选项都是合适的。**gamma** 和 **taub** 假定两个变量都是定序变量，而 **chi2**、**lrchi2** 和 **V** 则没有如此假定。

cchi2　　在二维表的每一个单元格内显示皮尔逊卡方。

cell 显示每一个单元格内的频数的总百分比。

chi2 对行变量和列变量独立的假设进行皮尔逊卡方检验。

clrchi2 显示一个二维表中每个单元格对似然比卡方的贡献。

column 显示每个单元格的列百分比。

exact 费舍的独立性精确检验。当表中有些单元格的期望频数很少时,此检验优于 **chi2**。但是在很大的表格中,这种检验运行太慢而显得不够实用。

expected 在假定独立性成立的情况下,二维表的每个单元格内的期望频数。

gamma Goodman 和 Kruskal 的等级相关系数 γ(gamma),并计算它的渐进标准误差(ASE)。这是基于同序对(concordant pairs)和异序对(discordant pairs)的数量(忽略同位秩,ties)对定序变量相关度的一种测量。值域为 $-1 \leqslant \gamma \leqslant 1$。

generate(new) 创造一组名为 *new1*、*new2* 等的虚拟变量以代表原来列表变量的取值。

lrchi2 对独立性假设的似然比卡方检验。如果表格中有任意的空单元格,将不能获得结果。

matcell(matname) 把报告的频数保存在变量 *matname* 中。[3]

matcol(matname) 把 $1 \times c$ 列的不重复数值保存在变量 *matname* 中。[4]

matrow(matname) 把 $r \times 1$ 行的不重复数值保存在 *matname* 中。[5]

missing 把缺失值也作为表的一行或一列。

nofreq 不显示单元格频数。

nokey 停止显示表格上方的表格注释(key)。只要单元格要计算一个以上统计量时,程序默认显示表格注释,否则将不显示。命令选项 **key** 将强制显示表格注释。

nolabel 显示数值而不是数值变量的值标签。

plot 对一维表提供简单的相对频数条形图。

replace 表示作为自变量被指定给 **tabi** 命令的即时数据将作为当前数据被保留在内存中,并且替代那里的任何数据。

row 显示每个单元格的行百分比。

sort 按照频数的降序方式显示行(如频数相等的话按变量的升序显示)。

subpop(*varname*) 在创建频数表时,排除那些 *varname* = 0 的观测值。相同的行和列由包括 *varname* = 0 组的所有数据来确定,因此表中可能会有一些频数是 0。

taub 输出 Kendall 的 τ_b(tau-b)及其渐近标准误(ASE)。这一指标测量两个定序变量之间的相关。**taub** 虽然与 **gamma** 类

---

[3]【译注:供编程使用。】

[4]【译注:供编程使用。】

[5]【译注:供编程使用。】

V 似,但是它对同位秩采取了一种修正。值域为 $-1 \leq \tau_b \leq 1$。

V Cramer 的 V(注意是大写)是一种名义变量关联度的测量。在 2×2 表中,有 $-1 \leq V \leq 1$。在更大的表中,值域为 $0 \leq V \leq 1$。

wrap 要求 Stata 不要为了增加可读性而对宽的二维表格做自动换行。除非特别指定了 **wrap**,通常出于可读性考虑,宽表会被分成几个小表。

对变量 *meetings* 和 *kids* 做交互表,取得列百分比(因为列变量 *kids* 是自变量)和卡方检验,键入:

. tabulate *meetings kids*, column chi2

```
+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

  Attended |
  meetings | Have children <19 in
        on |        town?
 pollution |        no        yes |     Total
-----------+----------------------+----------
        no |        52         54 |       106
           |     82.54      60.00 |     69.28
-----------+----------------------+----------
       yes |        11         36 |        47
           |     17.46      40.00 |     30.72
-----------+----------------------+----------
     Total |        63         90 |       153
           |    100.00     100.00 |    100.00

          Pearson chi2(1) =   8.8464   Pr = 0.003
```

从结果看,在有孩子的受访者中有 40% 参加了会议,而没有孩子的受访者中只有大约 17% 的人参加了会议。两变量的关联度统计性显著($P = 0.003$)。

偶尔我们可能需要在没有原始数据的情况下重新分析一个发表的表格。有一个专门的命令 **tabi**("直接"(immediate)做表)可以完成这项工作。在命令行上键入单元格频数,行之间用"\"隔开。这里,我们示范在只给定四个单元格频数的情况下如何用 **tabi** 做前面的卡方分析:

. tabi 52 54 \ 11 36, column chi2

```
+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

           |          col
       row |         1          2 |     Total
-----------+----------------------+----------
         1 |        52         54 |       106
           |     82.54      60.00 |     69.28
-----------+----------------------+----------
         2 |        11         36 |        47
           |     17.46      40.00 |     30.72
-----------+----------------------+----------
     Total |        63         90 |       153
           |    100.00     100.00 |    100.00

          Pearson chi2(1) =   8.8464   Pr = 0.003
```

不同于 **tabulate** 命令，**tabi** 并不需要引用内存中的任何数据。通过 **replace** 选项，我们要求 **tabi** 用新的交互表取代内存中的任何数据。在统计量的选项（**chi2**、**exact**、**nofreq** 等）上，**tabi** 与 **tabulate** 完全一样。

多表和多维交互表

对于调查和其他大型数据集，我们有时需要许多不同变量的频数分布。这时，我们可以分别做每个变量的频数分布，比如，先键入 **tabulate *meetings***，然后键入 **tabulate *gender*** 最后键入 **tabulate *kids***。除了这种方法之外，我们可以用另一个简单的专门命令 **tab1**：

```
. tab1 meetings gender kids
```

或者，要创建从 *gender* 直到 *school* 其间每个变量的频数表（一次最多可包括 30 个变量），键入：

```
. tab1 gender-school
```

同样，**tab2** 可以同时创建多个二维表。例如，下面的命令创建所列出变量的任何两个变量的二维交互表：

```
. tab2 meetings gender kids
```

tab1、**tab2** 与 **tabulate** 有相同的选项。

要创建多维列联表，一种方法是利用普通的 **tabulate** 加上一个前缀 **by**。下面是一个 *meetings* 与 *kids* 和 *contam*（受访者认为他或她的财产或水受到污染）的三维交互表，同时进行 *contam* 每一取值水平内 *meetings* 和 *kids* 的独立性卡方检验：

```
. by contam, sort: tabulate meetings kids, nofreq col chi2

-----------------------------------------------------------------
-> contam = no

  Attended |
  meetings | Have children <19 in
        on |         town?
 pollution |        no        yes |     Total
-----------+----------------------+----------
        no |     91.30      68.75 |     78.18
       yes |      8.70      31.25 |     21.82
-----------+----------------------+----------
     Total |    100.00     100.00 |    100.00

          Pearson chi2(1) =   7.9814   Pr = 0.005
-----------------------------------------------------------------

-> contam = yes

  Attended |
  meetings | Have children <19 in
        on |         town?
 pollution |        no        yes |     Total
-----------+----------------------+----------
        no |     58.82      38.46 |     46.51
       yes |     41.18      61.54 |     53.49
-----------+----------------------+----------
     Total |    100.00     100.00 |    100.00

          Pearson chi2(1) =   1.7131   Pr = 0.191
```

在受污染和未受污染的两个群体中，为父母者都更有可能参加会议。不过，只有在未受污染的较大群体中，"父母身份效应"才具有统计显著性。由于多维列联表把数据分成几个小的子样本，子样本的规模对显著性检验结果具有很大的影响。

这种方法可以扩展到更加复杂的交互表。例如,要创建一个 *gender* 、*contam* 、*meetings* 与 *kids* 之间的四维交互表,同时对每个 *meetings* 和 *kids* 的分表进行卡方检验,键入命令:

```
. by gender contam, sort:  tabulate meetings kids, column chi2
```

如果我们不需要百分比或者统计检验,还有一种更好的创建多维列联表的方法,就是利用 Stata 的一般表格创建命令 **table**。这个多功能命令有很多的选项,这里只示范其中的几个。要创建 *meetings* 的一个简单的频数分布表,键入:

```
. table meetings, contents(freq)

----------------------
Attended  |
meetings  |
on        |
pollution |      Freq.
----------+-----------
       no |        106
      yes |         47
----------------------
```

要创建一个二维频数表或交互表,键入:

```
. table meetings kids, contents(freq)

----------------------
          |   Have
Attended  | children
meetings  |  <19 in
on        |  town?
pollution |  no    yes
----------+-----------
       no |  52     54
      yes |  11     36
----------------------
```

如果我们列入第三个定类变量,它就会创建一个三维列联表的“大列”(supercolumns)。

```
. table meetings kids contam, contents(freq)

------------------------------------
          |       Believe own
          |     property/water
Attended  |  contaminated and Have
meetings  |  children <19 in town?
on        | --- no ---   --- yes --
pollution |  no    yes      no    yes
----------+-------------------------
       no |  42     44      10     10
      yes |   4     20       7     16
------------------------------------
```

更加复杂的列联表则需要 **by**( )选项,它允许有多达四维的“大行”(supperrow)变量。因此,**table** 可以创建多达七维的列联表:一行、一列、一大列以及多达四个大行。下面是一个四维的例子:

```
. table meetings kids contam, contents(freq) by(gender)

------------------------------------
Responden |
t's       |
gender    |       Believe own
and       |     property/water
Attended  |  contaminated and Have
meetings  |  children <19 in town?
on        | --- no ---   --- yes --
pollution |   no   yes      no   yes
----------+-------------------------
male      |
       no |   18    18       3     3
      yes |    2     7       3     6
----------+-------------------------
female    |
       no |   24    26       7     7
      yes |    2    13       4    10
------------------------------------
```

table 的 **contents**( )选项具体规定表格单元要包含什么统计量：

contents(freq)	频数
contents(mean *varname*)	*varname* 的平均数
contents(sd *varname*)	*varname* 的标准差
contents(sum *varname*)	*varname* 的总和
contents(rawsum *varname*)	忽略任意规定权数的总和
contents(count *varname*)	非缺失观测值的计数
contents(n *varname*)	等同于 **count**
contents(max *varname*)	*varname* 的最大值
contents(min *varname*)	*varname* 的最小值
contents(median *varname*)	*varname* 的中位数
contents(iqr *varname*)	*varname* 的四分位距
contents(p1 *varname*)	*varname* 的第 1 个百分位数
contents(p2 *varname*)	*varname* 的第 2 个百分位数

下一节将具体示范另外几个这种选项。

关于平均数、中位数以及其他概要统计指标的列表

tabulate 能够很容易地创建分类变量每一类别的平均数和标准差的列表。比如，要创建 *meetings* 每一类别内的 *lived* 平均值的一维表，键入：

```
. tabulate meetings, summ(lived)

   Attended |
meetings on |   Summary of Years lived in town
  pollution |        Mean   Std. Dev.       Freq.
------------+------------------------------------
         no |   21.509434   17.743809         106
        yes |   14.212766   13.911109          47
------------+------------------------------------
      Total |   19.267974   16.954663         153
```

从结果看，参加会议的人看来大都是新迁入者，平均在本地住了 14.2 年，而那些不参加会议的人则在本地平均住了 21.5 年。

我们也可以用 **tabulate** 创建一个关于平均值的二维表，键入：

```
. tabulate meetings kids, sum(lived) means

                    Means of Years lived in town

  Attended |
  meetings |   Have children <19
        on |       in town?
 pollution |        no        yes |     Total
-----------+----------------------+----------
        no | 28.307692  14.962963 | 21.509434
       yes | 23.363636  11.416667 | 14.212766
-----------+----------------------+----------
     Total | 27.444444  13.544444 | 19.267974
```

在参加会议的人当中,无论是做父母的人还是没做父母的人,在本地住的时间都相对比较短。因此,前面表中提到的新老住户的划分真实地反映了这样一个事实:有小孩的父母更可能参加会议。

上面用到的 **means** 选项要求表中只包括平均数。否则,我们将得到一个很庞大的表格,它包括平均数、标准差以及每个单元的频数。在第5章,我们将描述如何对分表的平均数做假设统计检验。

虽然 **table** 不能进行统计检验,但是它能很好地创建多达七维的包含平均数、标准差、总和、中位数或者其他统计量(请参见上一节所列的选项)的表格。这里有一个一维表格,它显示 *meetings* 每个类别下的 *lived* 平均值。

```
. table meetings, contents(mean lived)

-----------------------
Attended   |
meetings   |
on         |
pollution  | mean(lived)
-----------+-----------
        no |    21.5094
       yes |    14.2128
-----------------------
```

关于平均值的二维表格其实就是一种简单的扩展:

```
. table meetings kids, contents(mean lived)

----------------------------
Attended   |
meetings   |Have children <19
on         |     in town?
pollution  |      no       yes
-----------+----------------
        no | 28.3077    14.963
       yes | 23.3636   11.4167
----------------------------
```

表中的单元格也可以包含一个以上的统计量。假设我们想做一个二维表格,让它同时包含变量 *lived* 的平均数和中位数:

```
. table meetings kids, contents(mean lived median lived)

----------------------------
Attended   |
meetings   |Have children <19
on         |     in town?
pollution  |      no       yes
-----------+----------------
        no | 28.3077    14.963
           |    27.5      12.5
           |
       yes | 23.3636   11.4167
           |      21         6
----------------------------
```

上表内的中位数验证了我们前面根据平均数获得的结论：参加会议的人，无论是具有父母身份的人还是不具有父母身份的人，都比那些不参加会议的人在本镇居住的时间短。每个单元格内的中位数都小于平均数，这反映出变量 *lived* 是正偏态（平均数被几个居住时间很长的受访者拉大了）。

table 还能在单元格内显示两个或者更多变量的平均数、中位数、总和或其他概要统计。

使用频数权数

summarize、**tabulate**、**table** 以及其他相关命令都可以和表示重复观测值数目的频数权数（frequency weight）一起使用。例如，文件 *sextab2.dta* 包含了英国性行为调查的一些结果（Johnson et_al,1992）。很显然，它有 48 个观测值：

```
Contains data from C:\data\sextab2.dta
  obs:            48                          British sex survey (Johnson 92)
 vars:             4                          11 Jul 2005 18:05
 size:           432 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
age             byte   %8.0g       age        Age
gender          byte   %8.0g       gender     Gender
lifepart        byte   %8.0g       partners   # heterosex partners lifetime
count           int    %8.0g                  Number of individuals
-------------------------------------------------------------------------------
Sorted by:  age  lifepart  gender
```

变量 *count* 表示了每一种特征组合的案例数，因此这个小数据集实际上包含来自 18 000 万以上受访者的信息。例如，有 405 个受访者是男性，年龄为 16 ~24，到目前为止还没有异性伴侣。

```
. list in 1/5

     +-----------------------------------+
     |   age   gender   lifepart   count |
     |-----------------------------------|
  1. | 16-24     male       none     405 |
  2. | 16-24   female       none     465 |
  3. | 16-24     male        one     323 |
  4. | 16-24   female        one     606 |
  5. | 16-24     male        two     194 |
     +-----------------------------------+
```

我们可以利用 *count* 作为频数权数来创建一个 *lifepart* 与 *gender* 的交互表：

```
. tabulate lifepart gender [fw = count]

          # |
  heterosex |
   partners |        Gender
   lifetime |      male     female |     Total
------------+----------------------+----------
       none |       544        586 |      1130
        one |      1734       4146 |      5880
        two |       887       1777 |      2664
        3-4 |      1542       1908 |      3450
        5-9 |      1630       1364 |      2994
        10+ |      2048        708 |      2756
------------+----------------------+----------
      Total |      8385      10489 |     18874
```

通常的 **tabulate** 选项也能处理频数权数。这里做一个同样的交互表,但它显示列百分比而不是频数:

```
. tabulate lifepart gender [fweight = count], column nof

         # |
 heterosex |
  partners |        Gender
  lifetime |      male     female |     Total
-----------+----------------------+----------
      none |      6.49       5.59 |      5.99
       one |     20.68      39.53 |     31.15
       two |     10.58      16.94 |     14.11
       3-4 |     18.39      18.19 |     18.28
       5-9 |     19.44      13.00 |     15.86
       10+ |     24.42       6.75 |     14.60
-----------+----------------------+----------
     Total |    100.00     100.00 |    100.00
```

其他类型的权数,比如,概率权数或者分析权数不能用于 **tabulate** 命令,因为这些权数的意义对于这个命令的主要选项来说是不清楚的。

频数权数的另一种应用可以通过 **summarize** 来示范。文件 *college1.dta* 是从《巴伦袖珍入学指南》(*Barron's Compact Guide to Colleges*, 1992)当中随机抽取的一个样本,它包含了 11 所美国大学的信息:

```
Contains data from C:\data\college1.dta
  obs:            11                          Colleges sample 1 (Barron's 92)
 vars:             5                          11 Jul 2005 18:05
 size:           429 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
school          str28  %28s                   College or university
enroll          int    %8.0g                  Full-time students 1991
pctmale         byte   %8.0g                  Percent male 1991
msat            int    %8.0g                  Average math SAT
vsat            int    %8.0g                  Average verbal SAT
-------------------------------------------------------------------------------
Sorted by:
```

变量中包括 *msat*,它是每一所学校的平均的数学学习能力测验(math Scholastic Aptitude Test,缩写为"数学 SAT")分数。

```
. list school enroll msat

     +--------------------------------------------+
     |                      school   enroll   msat |
     |--------------------------------------------|
  1. |            Brown University     5550    680 |
  2. |                 U. Scranton     3821    554 |
  3. | U. North Carolina/Asheville     2035    540 |
  4. |           Claremont College      849    660 |
  5. |           DePaul University     6197    547 |
     |--------------------------------------------|
  6. |      Thomas Aquinas College      201    570 |
  7. |            Davidson College     1543    640 |
  8. |        U. Michigan/Dearborn     3541    485 |
  9. |        Mass. College of Art      961    482 |
 10. |             Oberlin College     2765    640 |
     |--------------------------------------------|
 11. |         American University     5228    587 |
     +--------------------------------------------+
```

我们可以很容易地取得这 11 所学校的 *msat* 的平均数,键入:

```
. summarize msat

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        msat |        11    580.4545    67.63189        482        680
```

这个表给每个学校的平均数学 **SAT** 分同样的权数。然而，迪保罗大学(DePaul University)的学生数是托玛斯·阿奎那斯学院(Thomas Aquinas College)的30倍。为了把在校学生数考虑在内，我们用变量 *enroll*(入学)进行加权：

```
. summarize msat [fweight = enroll]

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        msat |     32691     583.064    63.10665        482        680
```

如果键入

```
. summarize msat [freq = enroll]
```

也可以获得同样的结果。

与未加权获得的平均数不同，用在校学生数进行加权后获得的平均数等于这些大学32 691名学生(假定他们都参加了数学 SAT)的平均数。然而，需要注意的是，对标准差、最小值或最大值却不能这样说。除了平均数，大部分个体水平的统计量并不能简单地通过对已经汇总的数据进行加权来计算。因此，我们在使用权数的时候要谨慎。虽然它们可能在一个特定的分析中有意义，但是当需要做许多不同种类分析的时候，权数对于作为整体的数据来说就不一定有意义。

5 方差分析和其他比较方法

方差分析(analysis of variance, ANOVA)包含关于平均数差异假设检验的一整套方法。它应用范围广泛,既有比较 x 各个类别的 y 平均数这样的简单分析,也有涉及多个分类型和测量型的 x 变量这样更为复杂的情形。针对单个平均数(单样本)或者配对平均数(两个样本)的 t 检验是方差分析的基本形式。

基于秩的“非参数”(nonparametric)检验,包括符号(sign)检验、Mann-Whitney 秩和检验以及 Kruskal-Wallis 秩和检验,采取了另一种途径来比较多个分布。这些检验对变量的测度、分布形态和散布情况做了较弱的假定。因而,比起方差分析及其他“参数”(parametric)检验来说,它们在更宽松的条件下仍然有效。谨慎的分析者有时会同时使用参数检验和非参数检验,并检查二者是否指向类似的结论。当参数检验和非参数检验结果不一致的时候,就需要进一步探究。

anova 是本书将要介绍的第一种 Stata 模型拟合命令。和其他命令一样,它具有相当大的灵活性,包含了很多种模型。**anova** 可以拟合单因素(one-way)和多因素(N-way)的方差分析,也可以拟合平衡设计或非平衡设计的以及具有缺失单元值设计的协方差分析(analysis of covariance, ANCOVA)。它还可以拟合因子型(factorial),嵌套型(nested),混合型(mixed)的设计或者重复测量设计(repeated-measures designs)。在执行 **anova** 命令后,后续的 **predict** 命令可以计算预测值、不同类型的残差、各种标准误以及诊断性(diagnostic)统计量。另一个后续命令 **test** 用来获得用户指定的虚无假设检验结果。**predict** 和 **test** 命令与其他 Stata 模型拟合命令的工作方式很相似,如 **regress** 等(第 6 章)。

通过下面的菜单选项,可以完成本章描述的大部分操作:

Statistics-Summaries, tables, & tests-Classical tests of hypotheses
经典假设检验

Statistics-Summaries, tables, & tests-Nonparametric tests of hypotheses
非参数检验

Statistics-Summaries, ANOVA /MANOVA 方差分析与多元方差分析

Statistics-General Post-estimation-Obtain Predictions, residuals, etc.
取得预测值、残差等

Graphics-Overlaid twoway graphs 叠并双因素图形

命令示范

. **anova *y x1 x2***

执行双因素方差分析,检验 *y* 平均数在 *x1* 和 *x2* 两个分类变量的交互类别之间的差别。

. **anova *y x1 x2 x1*x2***

执行双因素因子方差分析,包括两个分类变量的主效应以及 *x1* 与 *x2* 之间的交互效应(*x1* * *x2*)。

. **anova *y x1 x2 x3 x1*x2 x1*x3 x2*x3 x1*x2*x3***

执行三因素因子方差分析,三阶交互效应(*x1* * *x2* * *x3*)、二阶交互效应和主效应都包括在内。

. **anova *reading curriculum/teacher|curriculum***

拟合嵌套模型,用以检验三种类型课程对学生阅读能力(*reading*)的影响。教师变量 *teacher* 嵌套在课程变量 *curriculum* 里面(**teacher |curriculum**),因为每一门课程都指派了几位不同的老师任教。《基础参考手册》(*Base Reference Manual*)提供了其他嵌套方差分析的例子,包括分块设计(split-plot design)。

. **anova *headache subject medication*, repeated(*medication*)**

拟合重复测量的方差分析模型,检验三类头疼药(*medication*)对受试者不同程度头疼(*headache*)的疗效。样本由 20 个经常头疼的受试者构成。在这个研究中,每一个受试者在不同时间分别服用这三种头疼药。

. **anova *y x1 x2 x3 x4 x2*x3*, continuous(*x3 x4*) regress**

执行四个自变量的协方差分析,其中两个自变量(*x1* 和 *x2*)是分类变量,另外两个自变量(*x3* 和 *x4*)是测量变量。交互效应(*x2* * *x3*)也包括在里面,结果以回归表形式输出、而不是默认的方差分析表。

. **kwallis *y*, by(*x*)**

执行 Kruskal-Wallis 方法来检验 *y* 是否在 *x* 的 *k* 个类别($k>2$)上有同样的秩分布。

. **oneway *y x***

执行单因素方差分析,检验在 *x* 不同类别上 *y* 的平均数是否存在差异。也可以用命令 **anova y x** 来完成同样的分析,但输出的表格不同。

. **oneway *y x*, tabulate scheffe**

执行单因素方差分析,输出中包括样本平均数表和 Scheffé 多重比较检验(Scheffé multiple-comparison test)的结果。

. **ranksum *y*, by(*x*)**

执行 Wilcoxon 秩和检验(Wilcoxon rank-sum test)(也称 Mann-Whitney 的 *U* 检验),虚无假设为 *y* 在二分变量 *x* 的每个类别上具有同样的秩分布。如果我们假定两个秩分布具有相同的形态,这也可看作是检验 *y* 的两个中位数是否相等。

. serrbar *ymean se x*, scale(2)

根据平均数的数据构建一个标准误条形图(standard-error-bar plot)。变量 *ymean* 代表 *y* 的分组平均数,*se* 代表标准误,*x* 则是分类变量 *x* 的取值。**scale(2)** 要求条形从每个平均数开始扩展至 ±2 倍标准误(默认设置为 ±1 倍标准误)。

. signrank *y1* = *y2*

执行 Wilcoxon 配对符号秩检验(Wilcoxon matched-pairs sign-rank test)以验证 *y1* 和 *y2* 的秩分布相同。也能用于检验 *y1* 的中位数是否等于某一常数,比如,23.4,那么键入命令 **signrank y1 =23.4** 即可。

. signtest *y1* = *y2*

检验 *y1* 和 *y2* 的中位数是否相等(假定为配对数据,也就是说,两个变量都是测量同一观察样本)。键入 **signtest y1 =5** 将执行符号检验,此处的虚无假设为 *y1* 的中位数等于 5。

. ttest *y* = 5

执行单样本 *t* 检验,虚无假设为 *y* 的总体平均数等于 5。

. ttest *y1* = *y2*

执行单样本(配对差异)*t* 检验,虚无假设为 *y1* 和 *y2* 的总体平均数相等。这条命令的默认形式假定数据是配对的。对于非配对数据(*y1* 和 *y2* 分别从两个独立样本中测量),须加上选项 **unpaired**。

. ttest *y*, by(*x*) unequal

执行两样本 *t* 检验,虚无假设为:对 *x* 的两类来说,*y* 的总体平均数都相等。这里不用假定各总体具有同样的方差。(如果没有 **unequal** 选项,则 **ttest** 假定同方差。)

单样本检验

单样本 *t* 检验有表面上看起来不同的两种应用:

①检验样本平均数 $\bar{y}$ 是否显著地不同于某一假设值 μ_0。

②检验同一套观察值中的两个变量 *y1* 和 *y2* 的平均数是否显著地不同。这等价于检验 *y1* 减去 *y2* 之后得到差值(difference score)的平均数是否等于零。

虽然第二项应用涉及两个而不是一个变量的信息,但这两项应用实质上使用同一个公式。

文件 *writing.dta* 中的数据是用来评估一门基于文字处理的大学写作课(Nash and Schwartz, 1987)。在学生修习这门课的前后都收集了一些测量值,比如,定时写作中完成的句子数。研究者想知道选课以后的测量值是否有所提高。

```
. describe

Contains data from C:\data\writing.dta
  obs:            24                          Nash and Schwartz (1987)
 vars:             9                          12 Jul 2005 10:16
 size:           312  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              byte   %8.0g       slbl       Student ID
preS            byte   %8.0g                  # of sentences (pre-test)
preP            byte   %8.0g                  # of paragraphs (pre-test)
preC            byte   %8.0g                  Coherence scale 0-2 (pre-test)
preE            byte   %8.0g                  Evidence scale 0-6 (pre-test)
postS           byte   %8.0g                  # of sentences (post-test)
postP           byte   %8.0g                  # of paragraphs (post-test)
postC           byte   %8.0g                  Coherence scale 0-2 (post-test)
postE           byte   %8.0g                  Evidence scale 0-6 (post-test)
-------------------------------------------------------------------------------
Sorted by:
```

假定我们知道学生在前些年平均能够完成10个句子。在检查数据 *writing.dta* 描述的学生是否通过这门课有进步之前,我们希望知道他们在课程开始时是否和早期学生在本质上相似。换句话说,他们的前测(*preS*)平均数是否显著地不同于早期学生的平均数(10)。要进行单样本的 t 检验,虚无假设为 $H_0: \mu = 10$,键入命令:

```
. ttest preS = 10

One-sample t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
    preS |      24    10.79167    .9402034    4.606037    8.846708    12.73663
------------------------------------------------------------------------------
Degrees of freedom: 23

                          Ho: mean(preS) = 10

     Ha: mean < 10           Ha: mean != 10            Ha: mean > 10
       t =   0.8420             t =   0.8420             t =   0.8420
   P < t =   0.7958       P > |t| =   0.4084         P > t =   0.2042
```

所标注的 P > t 意味着"获得较大 t 值的可能性",也就是指单侧检验概率。双侧检验概率则表示为 P > |t| = 0.408 4,它代表的是获得较大 t 的绝对值的概率。因为这一概率比较高,我们没有理由拒绝虚无假设 $H_0: \mu = 10$。请注意,**ttest** 自动提供了平均数的95%置信区间。我们也可以获得其他水平的置信区间,比如说90%,只要在命令后加上选项 **level(90)** 即可。

对于非参数型检验,比如说符号检验,采用二项分布来检验关于某一中位数的假设。例如,我们可以检验前测值(*preS*)的中位数是否等于10。**signtest** 的结果同样告诉我们,没有理由拒绝虚无假设。

```
. signtest preS = 10
```

```
Sign test

        sign |    observed    expected
-------------+------------------------
    positive |          12          11
    negative |          10          11
        zero |           2           2
-------------+------------------------
         all |          24          24

One-sided tests:
  Ho: median of preS - 10 = 0 vs.
  Ha: median of preS - 10 > 0
      Pr(#positive >= 12) =
         Binomial(n = 22, x >= 12, p = 0.5) =  0.4159

  Ho: median of preS - 10 = 0 vs.
  Ha: median of preS - 10 < 0
      Pr(#negative >= 10) =
         Binomial(n = 22, x >= 10, p = 0.5) =  0.7383

Two-sided test:
  Ho: median of preS - 10 = 0 vs.
  Ha: median of preS - 10 != 0
      Pr(#positive >= 12 or #negative >= 12) =
         min(1, 2*Binomial(n = 22, x >= 12, p = 0.5)) =  0.8318
```

与 **ttest** 相似,**signtest** 也包含右侧概率,左侧概率和双侧概率。但和 **ttest** 使用对称的 *t* 分布不同的是,**sightest** 使用二项分布,其左侧概率和右侧概率并不相等。在本例中,仅有双侧概率有意义,因为我们检验的是数据 *writing.dta* 中的学生是否不同于先前的学生。

接下来,我们通过检验"课程前后完成句子数的平均数(也即是说 *preS* 和 *postS*)相同"这一虚无假设来检查学生在课程中是否有进步。命令 **ttest** 可以实现这一目的,结果发现存在明显的进步。

```
. ttest postS = preS

Paired t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   postS |      24      26.375    1.693779    8.297787    22.87115    29.87885
    preS |      24    10.79167    .9402034    4.606037    8.846708    12.73663
---------+--------------------------------------------------------------------
    diff |      24    15.58333    1.383019    6.775382    12.72234    18.44433
------------------------------------------------------------------------------

                Ho: mean(postS - preS) = mean(diff) = 0

 Ha: mean(diff) < 0          Ha: mean(diff) != 0          Ha: mean(diff) > 0
      t =  11.2676                 t =  11.2676                 t =  11.2676
  P < t =   1.0000           P > |t| =   0.0000           P > t =   0.0000
```

由于我们期望不仅仅是前测和后测之间有所"不同",而是要有所"进步",因此单侧检验是恰当的。结果显示,单侧概率四舍五入并保留4位小数后等于0("0.000 0"实际上意味着 $P<0.000\ 05$)。学生的平均句子完成数确实有了显著提高。基于这一抽样,我们有95%的把握说句子完成数增加了12.7~18.4。

t 检验假定变量服从正态分布。这一假定通常不是十分关键,因为这些检验都比较稳健(robust)。但是,当非正态性涉及严重的特异值,或者说出现在小样本中,我们将检验中位数而不是平均数,并使用不做正态分布假定的非参数检验,这样会更可靠。例如,Wilcoxon 符号秩检验仅假定分布是对称和连续的。对这样的数据采用符号秩检

验，可以获得实质上和 **ttest** 相同的结论，即学生的句子完成数显著提高了。由于两种检验在结论上一致，我们可以更加有把握地作出断言。

```
. signrank postS = preS

Wilcoxon signed-rank test

        sign |      obs   sum ranks    expected
-------------+---------------------------------
    positive |       24         300         150
    negative |        0           0         150
        zero |        0           0           0
-------------+---------------------------------
         all |       24         300         300

unadjusted variance      1225.00
adjustment for ties        -1.63
adjustment for zeros        0.00
                      ----------
adjusted variance        1223.38

Ho: postS = preS
             z =   4.289
    Prob > |z| =   0.0000
```

两样本检验

本章其余部分的例子来自于 Ward 和 Ault（1990）对在校大学生的抽样调查（*student2.dta*）。

```
. describe

Contains data from C:\data\student2.dta
  obs:           243                          Student survey (Ward & Ault 1990)
 vars:            19                          12 Jul 2005 10:16
 size:         6561 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              int    %8.0g                  Student ID
year            byte   %8.0g       year       Year in college
age             byte   %8.0g                  Age at last birthday
gender          byte   %9.0g       s          Gender (male)
major           byte   %8.0g                  Student major
relig           byte   %8.0g       v4         Religious preference
drink           byte   %9.0g                  33-point drinking scale
gpa             float  %9.0g                  Grade Point Average
grades          byte   %8.0g       grades     Guessed grades this semester

belong          byte   %8.0g       belong     Belong to fraternity/sorority
live            byte   %8.0g       v10        Where do you live?
miles           byte   %8.0g                  How many miles from campus?
study           byte   %8.0g                  Avg. hours/week studying
athlete         byte   %8.0g       yes        Are you a varsity athlete?
employed        byte   %8.0g       yes        Are you employed?
allnight        byte   %8.0g       allnight   How often study all night?
ditch           byte   %8.0g       times      How many class/month ditched?
hsdrink         byte   %9.0g                  High school drinking scale
aggress         byte   %9.0g                  Aggressive behavior scale
-------------------------------------------------------------------------------
Sorted by:  id
```

大约有 19% 的学生参加了大学生联谊会（男生联谊会或女生联谊会，fraternity or sorority）：

```
. tabulate belong

 Belong to |
fraternity/ |
  sorority |      Freq.     Percent        Cum.
------------+-----------------------------------
    member |         47       19.34       19.34
 nonmember |        196       80.66      100.00
------------+-----------------------------------
     Total |        243      100.00
```

另一个变量 *drink* 用一个 33 级量表来测量学生喝酒频度和程度。校园传闻让人猜测大学生联谊会成员在饮酒行为上不同于其他学生。用箱线图比较了 *drink* 的中位数在会员和非会员之间的差别，而条形图则比较了这一变量的平均数，两种结论都与这种传闻是一致的。图 5.1 把这两种图叠并到了一个图中。

```
. graph box drink, over(belong) ylabel(0(5)35) saving(fig05_01a)
. graph bar (mean) drink, over(belong) ylabel(0(5)35) saving(fig05_01b)
. graph combine fig05_01a.gph fig05_01b.gph, col(2) iscale(1.05)
```

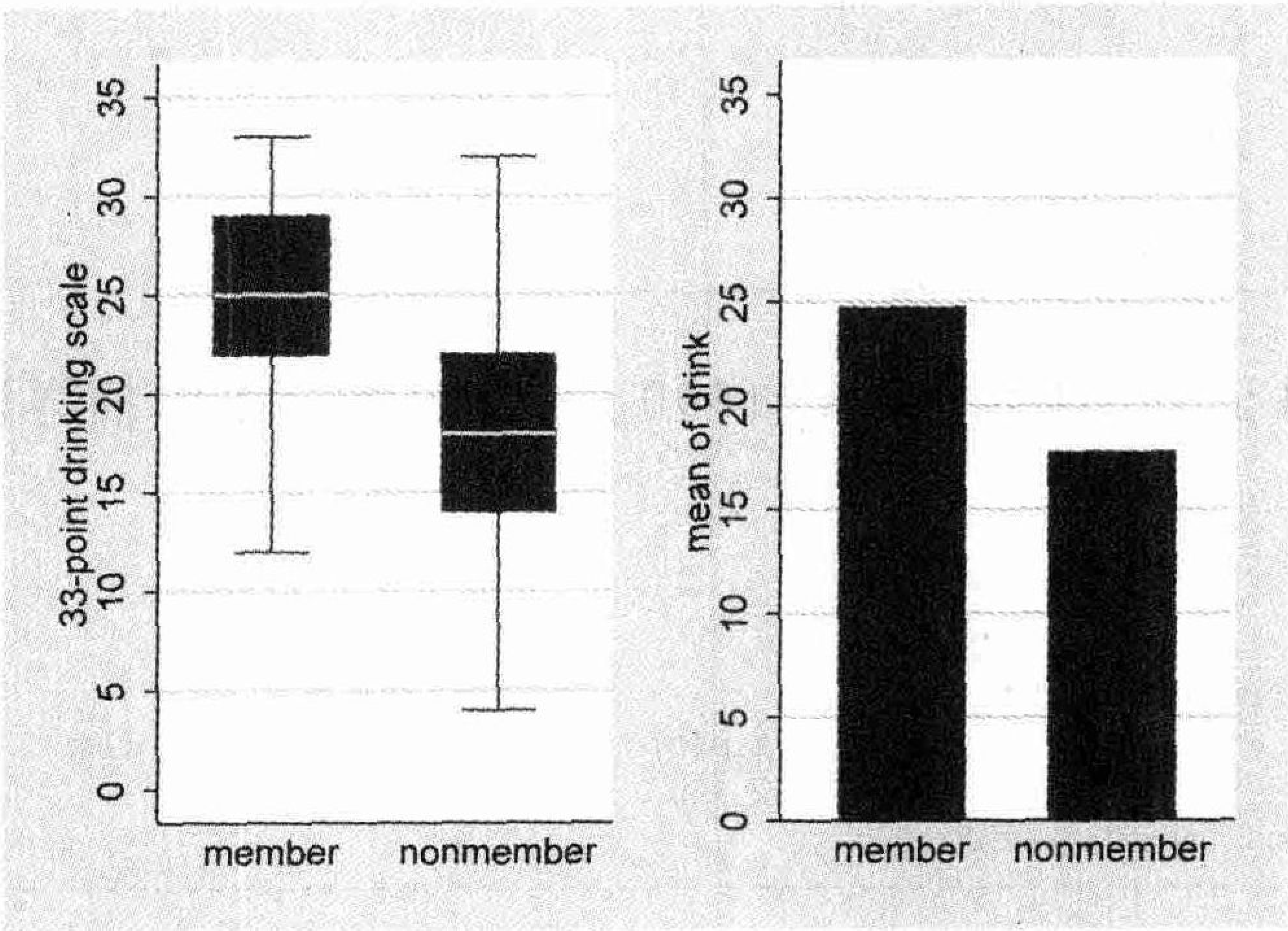


图 5.1

ttest 命令在前面用于单样本和配对差异检验，在这里同样可以用于两样本检验。在应用时，其一般命令语法是 **ttest *measurement*, by(*categorical*)** 。例如，

```
. ttest drink, by(belong)

Two-sample t test with equal variances

------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
  member |      47     24.7234    .7124518    4.884323    23.28931    26.1575
nonmembe |     196     17.7602    .4575013    6.405018    16.85792    18.66249
---------+--------------------------------------------------------------------
combined |     243      19.107     .431224    6.722117    18.25756    19.95643
---------+--------------------------------------------------------------------
    diff |              6.9632    .9978608                4.997558    8.928842
------------------------------------------------------------------------------
Degrees of freedom: 241

                 Ho: mean(member) - mean(nonmembe) = diff = 0

     Ha: diff < 0               Ha: diff != 0              Ha: diff > 0
       t =   6.9781              t =   6.9781              t =   6.9781
   P < t =   1.0000          P > |t| =   0.0000          P > t =   0.0000
```

输出结果表明,t 检验有赖于同方差假定。不过,大学生联谊会成员样本的标准差看起来要低一点。这意味着,比起非会员,他们所报告的饮酒行为更为相似。如果不假定同方差来执行类似的检验,在命令后加上选项 **unequal** 即可:

```
. ttest drink, by(belong) unequal

Two-sample t test with unequal variances

------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
  member |      47     24.7234    .7124518    4.884323    23.28931     26.1575
nonmembe |     196     17.7602    .4575013    6.405018    16.85792    18.66249
---------+--------------------------------------------------------------------
combined |     243      19.107     .431224    6.722117    18.25756    19.95643
---------+--------------------------------------------------------------------
    diff |              6.9632    .8466965                5.280627    8.645773
------------------------------------------------------------------------------
Satterthwaite's degrees of freedom:    88.22

                Ho: mean(member) - mean(nonmembe) = diff = 0

     Ha: diff < 0               Ha: diff != 0              Ha: diff > 0
       t =   8.2240               t =   8.2240               t =   8.2240
   P < t =   1.0000         P > |t| =   0.0000           P > t =   0.0000
```

对不等方差的校正并没有改变会员和非会员显著不同这一基本结论。我们可以进一步采用 Mann-Whitney 的 U 统计量非参数检验来检查这一结论,这一检验也称作 Wilcoxon 秩和检验。秩和检验假定秩分布具有相似的形状,此处的秩和检验结果表明我们可以拒绝不同总体中位数相等的假设。

```
. ranksum drink, by(belong)

Two-sample Wilcoxon rank-sum (Mann-Whitney) test

      belong |      obs    rank sum    expected
-------------+---------------------------------
      member |       47        8535        5734
   nonmember |      196       21111       23912
-------------+---------------------------------
    combined |      243       29646       29646

unadjusted variance   187310.67
adjustment for ties     -472.30
                     ----------
adjusted variance     186838.36

Ho: drink(belong==member) = drink(belong==nonmember)
             z =   6.480
    Prob > |z| =   0.0000
```

单因素方差分析

方差分析(ANOVA)提供比 t 检验更一般的方法来检验平均数之间的差异。最简单的例子是单因素(one-way)方差分析,它检验 y 的平均数是否在 x 的多个类别上都相等。单因素方差分析可以用 **oneway** 命令来执行,其一般形式是 **oneway *measurement categorical***(译者注:其中,*measurement* 代表测量型变量名,即 y;*categorical* 代表分类变量名,即 x)。例如,

```
. oneway drink belong, tabulate
```

```
  Belong to |
fraternity/ | Summary of 33-point drinking scale
   sorority |        Mean   Std. Dev.       Freq.
------------+------------------------------------
     member |   24.723404   4.8843233          47
  nonmember |   17.760204   6.4050179         196
------------+------------------------------------
      Total |   19.106996   6.7221166         243

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      1838.08426      1   1838.08426     48.69     0.0000
 Within groups      9097.13385    241   37.7474433
------------------------------------------------------------------------
    Total           10935.2181    242   45.1868517

Bartlett's test for equal variances:  chi2(1) =   4.8378  Prob>chi2 = 0.028
```

选项 **tabulate** 除了产生方差分析表本身外，还产生一个平均数和标准差表。针对二分变量 x 的单因素方差分析等价于两样本 t 检验，并且其 F 统计值等于相应的 t 值的平方。**oneway** 提供了更多的选项，运行也更快，但是缺乏 **ttest** 中的 **unequal** 选项，因此不能取消同方差假定。

oneway 运用 Bartlett 的卡方值来正规检验等方差假设。较低的 Bartlett 概率意味着方差分析中的同方差假定不大可能成立，在这种情况下，我们不应相信方差分析的 F 值检验结果。在上面的 **oneway *drink belong*** 示例中，Bartlett 的 $P=0.028$，于是令人质疑这一方差分析的有效性。

方差分析的真正价值并不在于两样本比较，而是对于三个或者更多平均数的复杂比较。例如，我们可以检验大学生饮酒行为平均数随着在校年数不同而变化的情况：

```
. oneway drink year, tabulate scheffe

    Year in | Summary of 33-point drinking scale
    college |        Mean   Std. Dev.       Freq.
------------+------------------------------------
   Freshman |      18.975   6.9226033          40
  Sophomore |   21.169231   6.5444853          65
     Junior |   19.453333   6.2866081          75
     Senior |   16.650794   6.6409257          63
------------+------------------------------------
      Total |   19.106996   6.7221166         243

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      666.200518      3   222.066839      5.17     0.0018
 Within groups      10269.0176    239   42.9666008
------------------------------------------------------------------------
    Total           10935.2181    242   45.1868517

Bartlett's test for equal variances:  chi2(3) =   0.5103  Prob>chi2 = 0.917

              Comparison of 33-point drinking scale by Year in college
                                    (Scheffe)
Row Mean-|
Col Mean |   Freshman   Sophomor     Junior
---------+---------------------------------
Sophomor |    2.19423
         |      0.429
         |
  Junior |    .478333    -1.7159
         |      0.987      0.498
         |
  Senior |   -2.32421   -4.51844   -2.80254
         |      0.382      0.002      0.103
```

我们可以拒绝平均数相等的虚无假设（P =0.001 8），但不能拒绝同方差假定（P = 0.917）。后者对于方差分析的有效性来说是个"好消息"。

下面的图 5.2 中的箱线图支持了这一结论，在每一个类别中都显示出相似的变化。这个图把箱线图和点图叠并在一起，显示出中位数和平均数的差异基本上遵从同样的模式。

```
. graph hbox drink, over(year) ylabel(0(5)35) saving(fig05_02a)
. graph dot (mean) drink, over(year) ylabel(0(5)35, grid)
      marker(1, msymbol(S)) saving(fig05_02b)
. graph combine fig05_02a.gph fig05_02b.gph, row(2) iscale(1.05)
```

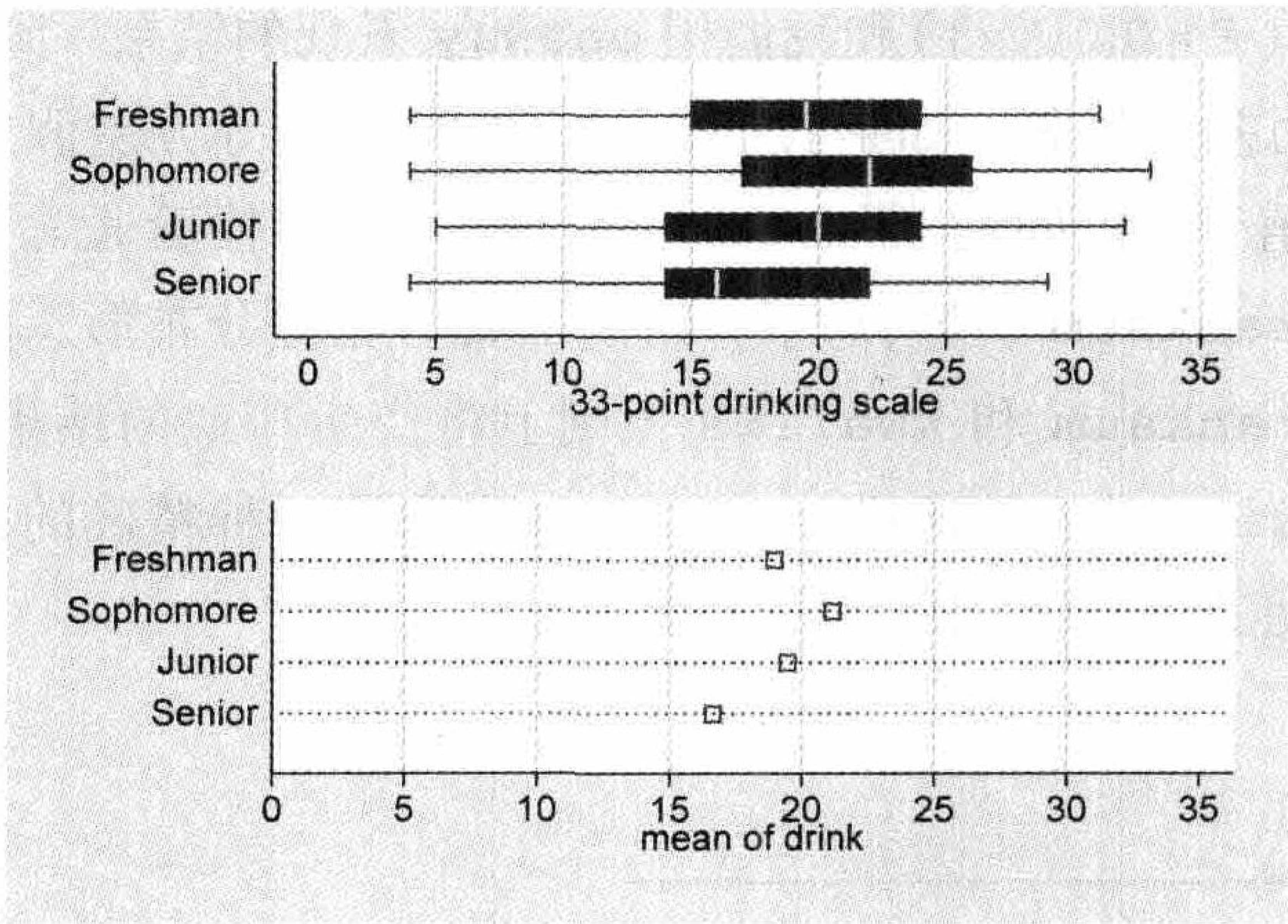


图 5.2

选项 **scheffe**（Scheffé 多重比较检验）生成了一个表来显示在每一对平均数之间的差异。一年级学生平均数等于 18.975，二年级学生平均数等于 21.169 23，因此，二年级和一年级学生之间的差值为 21.169 23 -18.975 =2.194 23，统计性上并非显著地不等于 0（P =0.429）。在表中 6 组对比中，只有四年级学生（Senior）和二年级学生（Sophomor）之间的差别是显著的（P =0.002），这一差别等于 16.6508 -21.169 2 = -4.518 44。因而，我们关于这个四个组平均数不同的总体结论主要来自于四年级学生（最轻度的饮酒者）和二年级学生（最重度的饮酒者）之间的对比。

oneway 提供了三种多重比较选项：**scheffe**、**bonferroni** 以及 **sidak**（其定义参见《基础参考手册》（*Base Reference Mannual*））。虽然 Scheffé 检验有时不够灵敏，但在大多数情况下都有效。

Kruskal-Wallis 检验（**kwallis**）是对两样本秩和检验在 K 个样本下的推广，它提供了一种非参数检验方法来代替单因素方差分析。它可以检验不同总体的中位数相等的虚无假设。

```
. kwallis drink, by(year)
```

```
Test: Equality of populations (Kruskal-Wallis test)

  +-----------------------------+
  |       year | Obs | Rank Sum |
  |------------+-----+----------|
  |   Freshman |  40 |  4914.00 |
  |  Sophomore |  65 |  9341.50 |
  |     Junior |  75 |  9300.50 |
  |     Senior |  63 |  6090.00 |
  +-----------------------------+

chi-squared =     14.453 with 3 d.f.
probability =      0.0023

chi-squared with ties =    14.490 with 3 d.f.
probability =      0.0023
```

在这里,**kwallis** 的分析结果($P=0.0023$)和我们用 **oneway** 发现的结果一致,即 *drink* 这一变量因在校年数不同而差异显著。如果我们怀疑方差分析的等方差假定或正态分布假定,或者我们怀疑特异值带来的问题,那么 Kruskal-Wallis 检验通常要比方差分析更为可靠。**kwallis** 和 **ranksum** 一样,对于各组内部具有相似分布形态的假定较弱。当应用于两样本分析时,**ranksum** 和 **kwallis** 原则上将产生相似的结果,但是这只有在数据中不包含同位秩(tie)时才是这样。**ranksum** 采取了一种精确的办法来处理同位秩的情况,这使得它在处理两样本问题时更为完美。

双因素和多因素方差分析

单因素方差分析检查测量型变量 y 在另一个分类变量 x 的不同类别之间的平均数差异。多因素方差分析推广这种探索,处理有两个或者更多 x 分类变量的情况。例如,我们可以考虑饮酒行为不仅仅因为是否联谊会成员而变化,同时也因为性别不同而变化。我们先来检查平均数的双因素表:

```
. table belong gender, contents(mean drink) row col

----------------------------------------
Belong to |
fraternit |
y/sororit |        Gender (male)
y         |   Female       Male     Total
----------+-----------------------------
   member | 22.44444   26.13793   24.7234
nonmember | 16.51724    19.5625   17.7602
          |
    Total | 17.31343   21.31193    19.107
----------------------------------------
```

在这个样本中,男性比女性饮酒要多,会员比非会员饮酒要多。会员—非会员之间的差别在男性和女性中类似。Stata 的多因素方差分析命令为 **anova**,它可以检验平均数之间的显著差异是否可以归诸于属于联谊会、性别以及这两者之间的交互作用(记为 *belong * gender*)。

```
. anova drink belong gender belong*gender
```

```
                 Number of obs =      243     R-squared     =  0.2221
                 Root MSE      = 5.96592     Adj R-squared =  0.2123

        Source |  Partial SS    df       MS           F     Prob > F
  -------------+----------------------------------------------------
         Model |  2428.67237     3  809.557456      22.75     0.0000
               |
        belong |   1406.2366     1   1406.2366      39.51     0.0000
        gender |  408.520097     1  408.520097      11.48     0.0008
 belong*gender |  3.78016612     1  3.78016612       0.11     0.7448
               |
      Residual |  8506.54574   239  35.5922416
  -------------+----------------------------------------------------
         Total |  10935.2181   242  45.1868517
```

在这个“双因素因子方差分析”的例子中,输出结果表明 *belong* 的主效应(P = 0.000 0)和 *gender* 的主效应(P =0.000 8)都是显著的,但它们的交互效应对模型贡献甚少(P =0.744 8)。这一交互效应并不显著地区别于0,因此我们可能更愿意拟合一个没有交互项,并且也更简单的模型(结果未显示)。

```
. anova drink belong gender
```

要在 **anova** 中纳入任何一项交互效应,只需指定有关变量名称,并用 * 号连接即可。除非各 x 的每种取值组合中的观察次数完全一样(“平衡数据”情形),否则很难在一个包含交互效应的模型中解释主效应的作用。当然,这并不是说在这样的模型中主效应不重要。正如在后面的章节所示,回归分析可以帮助理解复杂的方差分析结果。

协方差分析

协方差分析(analysis of covariance, ANCOVA)扩展了多因素方差分析,使之可以涵盖混合了分类变量和连续变量的情况。我们通过在 **anova** 命令中指明哪些变量是连续的来实现这一点。例如,当我们将 *gpa*(大学平均等级分)纳入自变量中,我们发现它也和饮酒行为有关。

```
. anova drink belong gender gpa, continuous(gpa)
```

```
                 Number of obs =      218     R-squared     =  0.2970
                 Root MSE      = 5.68939     Adj R-squared =  0.2872

      Source |  Partial SS    df       MS           F     Prob > F
  -----------+----------------------------------------------------
       Model |  2927.03087     3  975.676958      30.14     0.0000
             |
      belong |  1489.31999     1  1489.31999      46.01     0.0000
      gender |  405.137843     1  405.137843      12.52     0.0005
         gpa |    407.0089     1    407.0089      12.57     0.0005
             |
    Residual |  6926.99206   214  32.3691218
  -----------+----------------------------------------------------
       Total |  9854.02294   217  45.4102439
```

从这一分析我们知道,在我们控制了 *belong* 和 *gender* 之后,*drink* 和 *gpa* 之间还存在着显著性的关系。但除了用于统计显著性的 F 检验,方差分析或协方差分析通常并不提供很多变量之间如何联系的描述信息。回归分析(regression)凭借其清晰的模型和参数估计,在描述方面做得更好。因为方差分析和协方差分析可以算是回归分析的特例,我们可以用回归分析的形式重新表达它们。如果我们在 **anova** 命令中加上

regress选项,Stata 将自动完成这一过程。例如,我们可能希望看到回归的输出内容来帮助理解下列协方差分析结果。

```
. anova drink belong gender belong*gender gpa, continuous(gpa)
      regress

      Source |       SS       df       MS              Number of obs =     218
-------------+------------------------------           F(  4,   213) =   22.57
       Model |  2933.45823     4  733.364558           Prob > F      =  0.0000
    Residual |   6920.5647   213  32.4909141           R-squared     =  0.2977
-------------+------------------------------           Adj R-squared =  0.2845
       Total |  9854.02294   217  45.4102439           Root MSE      =  5.7001

------------------------------------------------------------------------------
       drink        Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
------------------------------------------------------------------------------
_cons            27.47676   2.439962    11.26   0.000      22.6672    32.28633
belong
          1      6.925384   1.286774     5.38   0.000     4.388942    9.461826
          2     (dropped)
gender
          1     -2.629057   .8917152    -2.95   0.004    -4.386774   -.8713407
          2     (dropped)
gpa             -3.054633   .8593498    -3.55   0.000    -4.748552   -1.360713
belong*gender
       1  1     -.8656158   1.946211    -0.44   0.657    -4.701916    2.970685
       1  2     (dropped)
       2  1     (dropped)
       2  2     (dropped)
------------------------------------------------------------------------------
```

加入 **regress** 选项,我们得到回归表格形式的 **anova** 输出结果。顶部是标准的方差分析表,给出了相同的整体 F 检验和 R^2。底部部分则描述了如下的回归分析:

我们构建了一个单独的虚拟变量(dummy variable),用{0,1}来表示每个 x 变量的每一个类别,但序次最高的类别被剔除,不包括在内。交互项(如果出现在变量列表中)则通过这些虚拟变量之间每一种可能组合的乘积来构建。将 y 对所有在命令行中指定的虚拟变量、交互项以及连续变量进行回归分析。

因此,前面的示例相当于对 *drink* 进行四个 x 变量的回归分析:

1.虚拟变量,编码为1 =联谊会成员,0 =其他情况(非会员作为 *belong* 最高序次的类别被删除了);

2.虚拟变量,编码为1 =女性,0 =其他情况(男性作为 *gender* 最高序次的类别被删除了);

3.连续变量,*gpa*;

4.交互项,编码为1 =女性联谊会会员,0 =其他情况。

这里要把单个虚拟变量的回归系数理解为它对 y 的预测值或者条件平均数的效应。例如,*gender* 的第一个类别(女性)的系数等于-2.629 057。这告诉我们,那些具有同样平均等级分(*gpa*)和相同会员身份的女性平均饮酒测量水平大约比同样情况的男性低2.63 个百分点。并且我们知道,对那些具有相同性别和相同会员身份的学生来说,*gpa* 每增长一分,平均饮酒测量值下降3.056 433。请注意,我们还得到了每个系数的置信区间和单项 t 检验结果,**anova, regress** 这一命令输出结果中所包含的信息比起单独的方差分析表要丰富得多。

预测值和误差条形图

在运行 **anova** 后,后续命令 **predict** 可以计算出预测值、残差、标准误以及各种诊

断统计量。这些统计量的应用之一是用误差条形图(error-bar chart)来画出模型的预测情况。举一个简单的例子,让我们回到 *drink* 对 *year* 的单因素方差分析。

```
. anova drink year

                   Number of obs =     243     R-squared     =  0.0609
                   Root MSE      = 6.55489     Adj R-squared =  0.0491

      Source |  Partial SS    df       MS           F     Prob > F
  -----------+----------------------------------------------------
       Model |  666.200518     3  222.066839      5.17     0.0018
             |
        year |  666.200518     3  222.066839      5.17     0.0018
             |
    Residual |  10269.0176   239  42.9666008
  -----------+----------------------------------------------------
       Total |  10935.2181   242  45.1868517
```

为了从最近的 **anova** 运行结果基础上计算预测值,键入 **predict** ,后面接上新变量的名称:

```
. predict drinkmean
(option xb assumed; fitted values)

. label variable drinkmean "Mean drinking scale"
```

加上 **stdp** 选项,**predict** 将计算预测平均数的标准误。

```
. predict SEdrink, stdp
```

使用这些新的变量,我们应用 **serrbar** 命令来创建一个误差条形图。选项 **scale(2)** 告诉 **serrbar** 画出一个正负 2 倍标准误的条形图,从

drinkmean $-2 \times$ *SEdrink*

到

drinkmean $+2 \times$ *SEdrink*

在 **serrbar** 命令中,最先列出的变量应该是平均数或称 y 变量。接下来列出的是标准误或者标准差(取决于你想显示哪一种);第三个列出的变量用来定义 x 轴。**serrbar** 中的 **plot( )** 选项可以指定第二个图,并使其叠并显示在标准的误差条形图上。在图 5.3 中,我们叠并显示了一个折线图,它用实线段将 *drinkmean* 的各个取值连接起来。

```
. serrbar drinkmean SEdrink year, scale(2)
     plot(line drinkmean year, clpattern(solid)) legend(off)
```

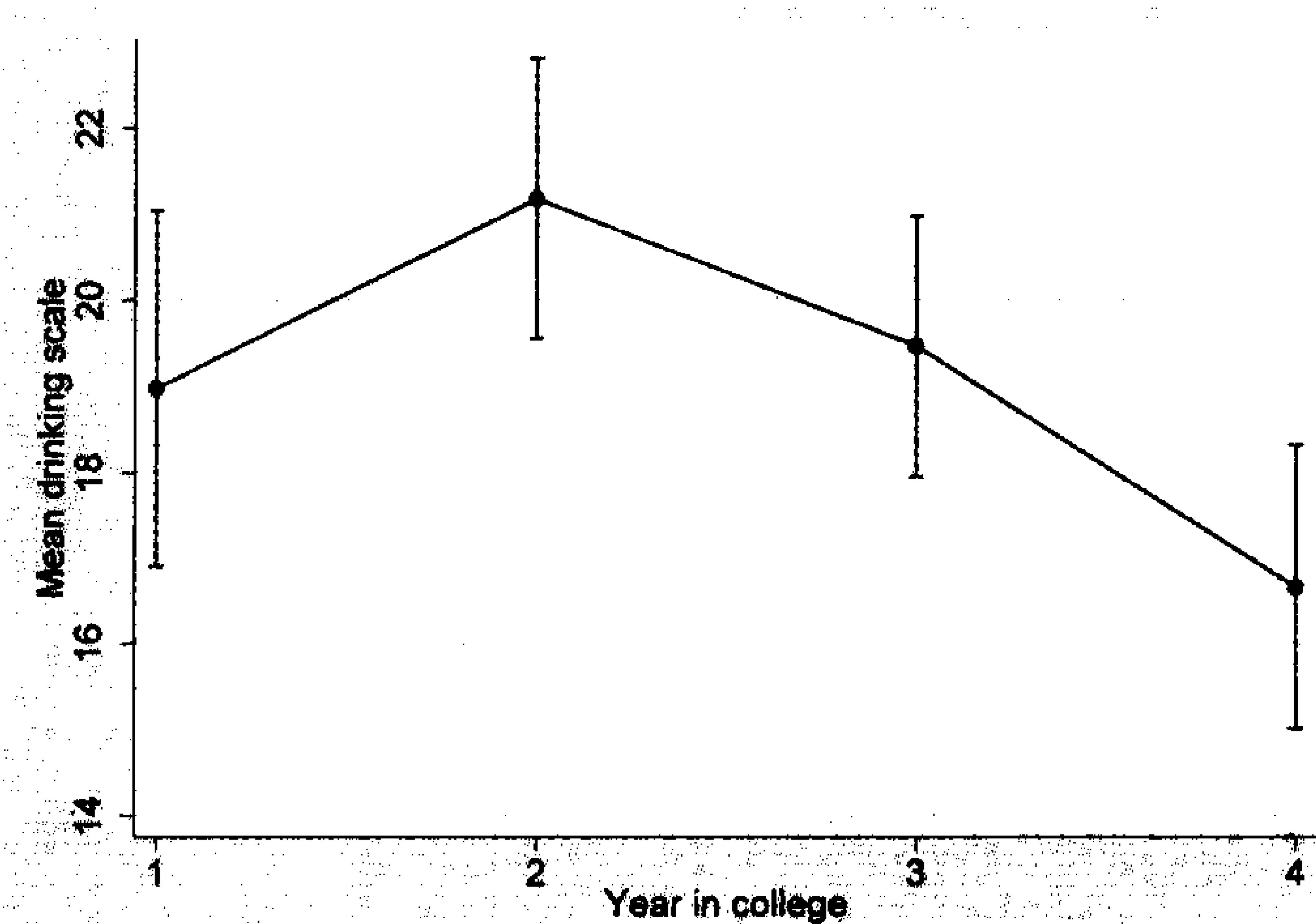


图 5.3

对一个双因素因子方差分析来说,误差条形图可以帮助我们将主效应和交互效应可视化。虽然常用的误差条形图命令 **serrbar** 经过努力可以达到这一目的,但使用 **graph twoway** 这类命令来制图显得更加灵活,下面我们进行示范。首先,我们执行方差分析,获得组平均数(即预测值)和它们的标准误,接下来创建新变量,令其值等于组平均数加上或者减去两倍的标准误。这个例子用来检测学生的攻击性行为(*aggress*)与性别(*gender*)以及在校年数(*year*)之间的关系。性别、在校年数以及它们之间的交互效应都表现出统计性显著。

```
. anova aggress gender year gender*year

                    Number of obs =      243     R-squared     =  0.2503
                    Root MSE      = 1.45652     Adj R-squared =  0.2280

        Source |  Partial SS    df       MS           F     Prob > F
   ------------+----------------------------------------------------
         Model |  166.482503     7  23.7832147      11.21     0.0000
               |
        gender |  94.3505972     1  94.3505972      44.47     0.0000
          year |  19.0404045     3  6.34680149       2.99     0.0317
   gender*year |  24.1029759     3  8.03432529       3.79     0.0111
               |
      Residual |  498.538073   235  2.12143861
   ------------+----------------------------------------------------
         Total |  665.020576   242  2.74801891

. predict aggmean
(option xb assumed; fitted values)

. label variable aggmean "Mean aggressive behavior scale"

. predict SEagg, stdp

. gen agghigh = aggmean + 2 * SEagg

. gen agglow = aggmean - 2 * SEagg

. graph twoway connected aggmean year
      ||  rcap agghigh agglow year
      ||  , by(gender, legend(off) note(""))
      ytitle("Mean aggressive behavior scale")
```

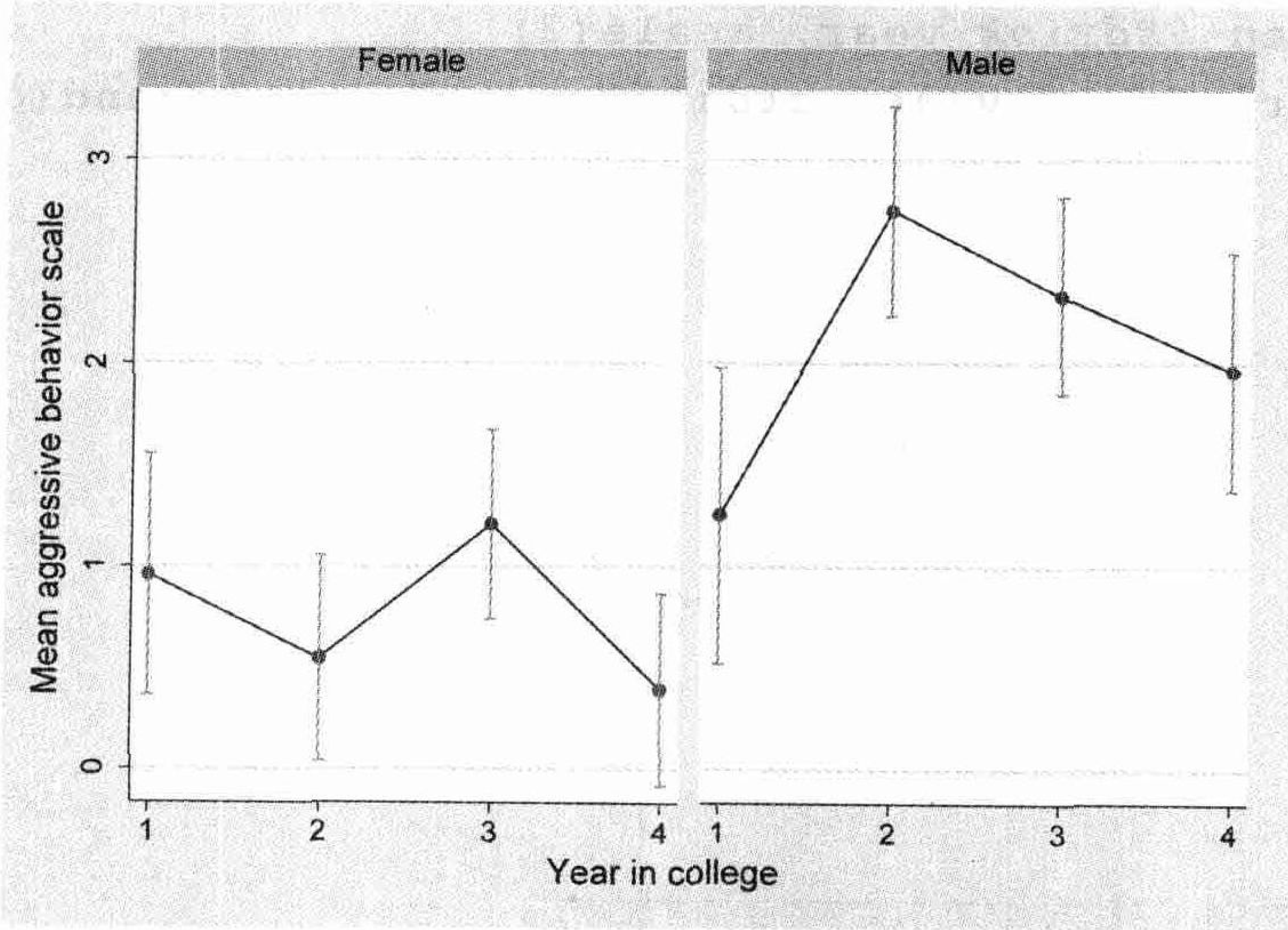


图 5.4

图 5.4 通过叠并两对图得到了误差条形图。第一对是分别女生和男生的折线图，它把 *aggress* 的组平均数做了连线（我们用 **predict** 来计算平均数，并存为变量 *aggmean*）。第二对是女生和男生的戴帽芒线图（**twoway rcap**），垂直方向的芒线将变量 *agghigh*（*aggress* 的组平均数加上两倍标准误）和 *agglow*（*aggress* 的组平均数减去两倍标准误）连接起来。选项 **by(gender)** 用来产生女生和男生分图。注意，在一幅使用 **by( )** 选项的图中，为了不让产生图例和注释，**legend(off)** 和 **note(" ")** 必须作为子选项出现在选项 **by( )** 的括号中。

最后输出的误差条形图（图 5.4）显示，女性在攻击行为测量上的平均数在大学四年中都是在低水平波动。男性的平均数则始终比较高，并且在二年级达到顶峰，这一点和前面讨论的饮酒情况相似（图 5.2 和图 5.3）。因此，*aggress* 和 *year* 之间的关系在男性和女性之间是不同的。这张图有助于我们理解和解释显著的交互效应。

predict 在回归分析（ **regress** ）和方差分析（ **anova** ）中的应用方式完全一样，这是因为这两种方法采用了同样的数学分析框架。在第 6 章中列出了 **predict** 命令的其他选项，在第 7 章中则给出了更多如何应用这些选项的例子。这些选项包括残差，它们可以用来检验误差分布的假定；还包括一套诊断性统计量（比如，杠杆作用、Cook 的 *D* 统计量以及 *DFBETA*），它们测量单个观测值对模型结果的影响。在第 13 章中描述的 Durbin-Watson 检验（ **dwstat** ）可以在执行 **anova** 之后检验一阶自相关的情况。条件效应标绘图则提供了绘图工具，用来理解更为复杂的回归、方差分析或协方差分析模型。

6 线性回归分析

Stata 提供了范围异常宽广的回归程序。键入命令 **help regress** 可以看到这些应用的部分清单。本章介绍 **regress** 及其相关命令，它们完成各种简单和多元(multiple)常规最小二乘法(ordinary least squares,OLS)回归。其后续命令 **predict** 可以计算预测值、残差以及诸如杠杆作用或者 Cook 的 *D* 之类的诊断性统计量。另一后续命令 **test** 检验用户指定的假设。**regress** 可以完成包括加权最小二乘法回归和两阶段最小二乘法在内的其他分析。涉及虚拟变量、交互效应、多项式和逐步变量筛选的回归分析在本章中也简要地有所提及，同时对残差分析给予了初步介绍。

通过下面的菜单选择可以实现绝大部分讨论到的操作：

Statistics-Linear regression and related-Linear regression 线性回归

Statistics-Linear regression and related-Regression diagnostics 回归诊断

Statistics-Post-estimation-Predictions, residuals, etc 取得预测值、残差等

Graphics-Overlaid twoway graphs 叠并二维图

Statistics-Longitudinal /panel data 纵贯及面板数据

命令示范

. **regress *y x***

执行 *y* 对单个预测变量 x 的常规最小二乘法(OLS)回归。

. **regress *y x* if *ethnic* == 3 & *income* > 50**

执行 *y* 对 *x* 的回归，但只使用 *ethnic* 等于 3 并且 *income* 大于 50 的数据子集。

. **predict *yhat***

创建一个新变量(此处任意命名为 *yhat*)，并令其等于最近回归所得到的预测值。

. **predict *e*, resid**

创建一个新变量(此处任意命名为 *e*)，并令其等于最近回归所得到的残差。

. **graph twoway lfit *y x* || scatter *y x***

画出 *y* 相对于 *x* 的散点图(scatterplot)，并加入简单回归线(**lfit**，即线性拟合)。

. graph twoway mspline *yhat x* || scatter *y x*

在 *y* 对 *x* 的散点图上加画简单回归线,(用修匀的立方样条曲线)连接回归预测值(本例中变量名为 *yhat*)。

注意:在 Stata 中画回归直线或者曲线时有很多选择。它们包括 **twoway** 制图类型的 **mspline** (如上所示)、**mband**、**line**、**lfit**、**lfitci**、**qfit** 和 **qfitci** ,并且每一种都有其自身的优点和不同选项。通常,我们将直线或者曲线和散点图组合(叠并)起来。如果像上面的示例那样,散点图在我们的 **graph twoway** 命令中排在第二位,散点将打印在回归线的上层。如果在命令中先写散点图命令,那么回归线将打印在散点的上层。在本章和后面章节中的示例阐明了这些不同的可能性。

. rvfplot

自动采用最近的回归分析结果画出残差对拟合值(即预测值)的标绘图(residual versus fitted plot)。

. graph twoway scatter *e yhat*, yline(0)

用变量 *e* 和 *yhat* 画出残差对预测值的标绘图(residual versus predicted values plot)。

. regress *y x1 x2 x3*

执行 *y* 对 *x1*、*x2*、*x3* 这三个预测变量的多元回归。

. regress *y x1 x2 x3*, robust

计算稳健标准误的(Huber/White 法)估计值。详情参见《用户指南》(*User's Guide*)。**robust** 选项在许多其他模型的拟合中也同样可以应用。

. regress *y x1 x2 x3*, beta

执行多元回归,并且将标准化回归系数(又称"β 权数",beta weights)包含在输出表中。

. correlate *x1 x2 x3 y*

显示皮尔逊相关矩阵(matrix of Pearson correlation),使用的数据仅包括那些在所有指定变量均无缺失值的案例。如果加上 **covariance** 选项,将输出方差协方差矩阵来取代相关矩阵。

. pwcorr *x1 x2 x3 y*, sig

显示皮尔逊相关矩阵,采用配对删除(pairwise deletion)的方法去掉缺失值,并显示对每个相关系数进行 H_0: $\rho = 0$ 的 *t* 检验概率。

. graph matrix *x1 x2 x3 y*, half

画散点图矩阵。由于变量列表相同,这一示例产生的矩阵和前面的 **pwcorr** 命令产生的相关矩阵具有同样的排列方式。因变量 *y* 列在最后面,这使得最底部的一行形成一系列 *y* 对 *x* 之间的标绘图。

. test *x1 x2*

执行 *F* 检验,虚无假设为:在最近的回归模型中,*x1* 和 *x2* 的系数都等于零。

.xi: regress *y x1 x2* i.*catvarx2***

执行对 y 的"扩展交互效应"回归,预测变量为 x1、x2 和一套自动生成并代表分类变量 catva 各类别的虚拟变量以及一套交互项,交互项等于虚拟变量和测量型变量 x2 的乘积。**help xi** 给出了有关详情。

```
. sw regress y x1 x2 x3, pr(.05)
```

执行反向剔除的逐步回归,直到所有保留的预测变量都在 0.05 水平上显著。所有列表上的预测变量都进入第一次迭代,此后每一步迭代去掉 P 值最高的预测变量,直到所有保留下来的预测变量的概率值在"保留概率"以下为止,这就是 **pr(.05)** 选项的意义。其他选项允许正向纳入法或者分层选择。对于许多其他模型拟合命令来说,都存在着不同的逐步分析方式。键入 **help sw** 可以得到一个清单。

```
. regress y x1 x2 x3 [aweight = w]
```

执行 y 对 x1、x2 和 x3 的加权最小二乘回归(WLS)。变量 w 代表分析权数(analytical weight),这种加权相当于我们将每一个变量和常数都乘以 w 的平方根、然后所执行的常规回归。当 y 变量和 x 变量为平均数、率或者比例,并且 w 是每一汇总观察(如城市或学校)中的个体数目时,分析权数通常用来校正异方差性(heteroskedasticity)。当 y 和 x 为个体层次变量,而权数表示重复测量时,则应该采用频数权数选项来加权,即[**fweight = w**]。当权数表示设计因子(design factors),比如,非等比抽样,请参见 **help svy**。

```
. regress y1 y2 x (x z)
. regress y2 y1 z (x z)
```

使用工具性变量 x 和 z 估计 y1 和 y2 之间的相互作用,第一部分命令定义了结构方程[6]:

$$y1 = \alpha_0 + \alpha_1 y2 + \alpha_2 x + \varepsilon_1$$
$$y2 = \beta_0 + \beta_1 y1 + \beta_2 z + \varepsilon_2$$

命令中的括号包含了在所有方程中的外生(exogenous)变量。在这个例子中,**regress** 完成了两阶段最小二乘(two-stage least squares,2SLS)分析。

```
. svy: regress y x1 x2 x3
```

对 y 就预测变量 x1、x2 和 x3 进行回归,并对复杂抽样设计进行恰当的调整。我们假定先前曾用 **svyset** 命令设置过数据,已经指定分层、群集和抽样概率。键入 **help svy** 可列出能用于复杂抽样调查数据的多种程序。**help regress** 概括描述了这一单独命令的程序语法;更详细的内容请参考《用户指南》和《调查数据参考手册》(*Survey Data Reference Manual*)。

```
. xtreg y x1 x2 x3 x4, re
```

用一般化最小二乘法(GLS)可拟合具有随机效应的面板(即横剖时间序列,cross-sectional time series)模型。一条面板数据中的观察值由第 i 个单位在时间 t 的信息组成,并且每一个单位都有多次观察。在使用命令 **xtreg** 之前,用来确定单位的变量由命令 **iis**(表示"i 是",i is)来指定,用来确定时间的变量则用命令 **tis**(即"t 是")来指定。一旦数据被存盘,这些定义就被保留下来,供 **xtreg** 和其他 **xt** 程序在将来进行分析。**help xt** 可以列出可用的面板估计程序。**help xtreg** 给出这一命令的程序语法以及印刷版文档的参考信息。如果所用数据的每个单位都包括很多观

---

[6]【译注:翻译中由原作者对下面公式作了修改。】

察,采用时间序列方法就更为合适。Stata 的时间序列程序(在第 13 章中介绍)提供了分析面板数据的进一步工具。详尽的描述请参考《纵向/面板数据参考手册》(*Longitudinal/Panel Data Reference Manual*)。

. **xtmixed *population year* || *city*: *year***

假定有不同城市每年的人口数据。和常规回归相似,命令中的 **xtmixed *population year*** 部分定义了"固定效应"(fixed-effect)模型,该模型描述人口的平均变动趋势。命令中的 || ***city*: *year*** 部分则定义了"随机效应"(random-effect)模型,允许每一个城市具有特定的截距和斜率(即不同的起点和增长率)。

. **xtmixed *SAT grades prepcourse* || *region*:|| *district*: *pctcollege***

拟合一个分层的(即嵌套的或多层的)线性模型,并将学生的 SAT 成绩作为如下因素的函数进行预测[7]:个体学生的年级(*grades*)以及是否参加了某门预备课程(*prepcourse*);该校所在的地区(*region*,它仅仅影响 *y* 的截距);还有本校区(*district*)成年人中的大学毕业生比例(*pctcollege*)。欲知其更多详情,请参见《纵向/面板数据参考手册》中关于命令 **xtmixed** 的内容。

回归表

文件 *states.dta* 包含了美国各州以及哥伦比亚地区的教育数据。

. **describe *state csat expense percent income high college region***

```
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
state           str20  %20s                   State
csat            int    %9.0g                  Mean composite SAT score
expense         int    %9.0g                  Per pupil expenditures prim&sec
percent         byte   %9.0g                  % HS graduates taking SAT
income          long   %10.0g                 Median household income
high            float  %9.0g                  % adults HS diploma
college         float  %9.0g                  % adults college degree
region          byte   %9.0g       region     Geographical region
```

政治领导人偶尔使用学术能力测试(Scholastic Aptitude Test,SAT)的平均分数对美国各州教育体系进行有针对性的评价。例如,有人提出教育支出多的州是否SAT 成绩也较高的问题。我们可以通过对平均综合 SAT 成绩(*csat*)就学生人均支出(*expense*)进行回归。适当的 Stata 命令形式为 **regress *y x***,这里 *y* 是被预测的变量或者说是因变量,而 *x* 则是预测变量或自变量。

. **regress *csat expense***

```
      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  1,    49) =   13.61
       Model |  48708.3001     1  48708.3001           Prob > F      =  0.0006
    Residual |   175306.21    49  3577.67775           R-squared     =  0.2174
-------------+------------------------------           Adj R-squared =  0.2015
       Total |   224014.51    50  4480.2902            Root MSE      =  59.814

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |  -.0222756   .0060371    -3.69   0.001    -.0344077   -.0101436
       _cons |   1060.732    32.7009    32.44   0.000     995.0175    1126.447
------------------------------------------------------------------------------
```

[7]【译注:在翻译过程中,此命令及有关解释由原作者进行了修改。】

这一回归结果颇为令人意外：州教育开支越多，学生的平均 SAT 值反而越低。虽然任何因果解释在此时为时尚早，但回归表确实传递了 *csat* 和 *expense* 之间的线性统计关系。基于表顶部左边的离差平方和，表顶部的右边给出了整体 *F* 检验结果。*F* 检验用来评估关于模型中所有 *x* 变量（此处只有一个 *x* 变量，*expense*）的系数都等于 0 的虚无假设。*F* 统计量等于 13.61，其自由度为 1 和 49，很容易地导致拒绝这一虚无假设（*P* =0.000 6）。Prob > F 指的是，在虚无假设为真的情况下，我们从这样一个总体中随机抽取样本时"出现较大 *F* 值的可能性"。

在顶部右端，我们还看到确定系数（coefficient of determination），R^2 = 0.217 4。学生人均支出解释了州际平均综合 SAT 成绩差异的 22%，调整的 R^2，即 R_a^2，考虑了由数据复杂性带来的模型复杂性，于是有 R_a^2 = 0.201 5。这一调整统计量往往为研究提供了更为有用的信息。

回归表的下半部给出了拟合模型。我们在第一栏看到回归系数（斜率和 y 的截距），由此得到预测方程：

预测值 *csat* = 1 060.732 - 0.222 756 *expense*

第二栏列出了系数的估计标准误。它们用来进行 *t* 检验（第 3 ~4 栏），并计算每一回归系数的置信区间（第 5 ~6 栏）。*t* 值（各系数除以它们各自的标准误）检验总体中相应系数等于 0 的虚无假设。在 α = 0.05 的显著水平下，根据 *expense* 的系数（*P* = 0.001）和 *y* 截距的系数（"0.000"的结果意味着 $P < 0.0005$），我们可以拒绝虚无假设。Stata 的模型命令例行地输出 95% 的置信区间，但我们可以通过指定 **level( )** 选项来要求其他水平的置信区间，如下所示：

```
. regress csat expense, level(99)
```

由于这些数据并不代表从美国各州大总体中抽出的随机样本，因此假设检验和置信区间都不具备它们的通常意义。本章对此讨论不过是出于示范的目的。

_cons 这一项代表回归常数，通常设为 1。除非我们告诉 Stata 不要常数项，否则它将自动包括一个常数项。选项 **nocons** 将使 Stata 取消常数项，并执行通过原点的回归。例如，

```
. regress y x,  nocons
```

或者

```
. regress y x1 x2 x3, nocons
```

在某些高级应用中，用户需要指定自己所设的常数项。如果"自变量"中包括用户补充的常数项（例如，取名叫 *c*），那就采用 **hascons** 选项来代替 **nocons**：

```
. regress y c x, hascons
```

在这种情况下使用 **nocons** 将得到一个令人误解的 *F* 检验和 R^2。更多关于 **hascons** 的内容请参考《基础参考手册》，或者使用 **help regress** 命令咨询。

多元回归

多元回归允许我们在考虑到其他一些预测变量的同时估计 *expense* 是如何预测 *csat* 的。我们可以通过简单地将那些自变量列在命令中来引入 *csat* 的其他预测变量。

```
. regress csat expense percent income high college

      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  5,    45) =   42.23
       Model |  184663.309     5  36932.6617           Prob > F      =  0.0000
    Residual |  39351.2012    45  874.471137           R-squared     =  0.8243
-------------+------------------------------           Adj R-squared =  0.8048
       Total |   224014.51    50   4480.2902           Root MSE      =  29.571

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |   .0033528   .0044709     0.75   0.457     -.005652    .0123576
     percent |  -2.618177   .2538491   -10.31   0.000    -3.129455   -2.106898
      income |   .0001056   .0011661     0.09   0.928     -.002243    .0024542
        high |   1.630841    .992247     1.64   0.107     -.367647    3.629329
     college |   2.030894   1.660118     1.22   0.228    -1.312756    5.374544
       _cons |   851.5649   59.29228    14.36   0.000     732.1441    970.9857
------------------------------------------------------------------------------
```

由此得到多元回归方程：

预测值 *csat* =851.56 +0.003 35*expense* −2.618*percent* +0.000 1*income* + 1.63*high* +2.03*college*

控制其他四个变量削弱了 *expense* 的影响，由 −0.022 3 变为 0.003 35，这一系数不再显著地区别于 0。在先前的简单回归中所发现的 *expense* 和 *csat* 之间令人意外的负向关系显然可以被其他预测变量解释。

只有 *percent*（参加 SAT 考试的高中毕业生比例）的系数在 0.05 的水平下显著。我们可以按照如下方式理解这一"四阶偏回归系数（fourth-order partial regression coefficient）"（之所以这样说是因为它的计算按四个其他自变量做过调整）。

b2 = −2.618：在 *expense*、*income*、*high* 和 *college* 保持不变的情况下，参加 SAT 考试的高中毕业生比例每增加 1 个百分点，预测的 SAT 平均分数下降 2.618 分。

总的来说，模型中的五个变量解释了 80% 的平均综合 SAT 成绩的方差（R_a^2 = 0.804 8）。相比之下，我们早期用 *expense* 作为唯一预测变量的简单回归仅仅解释了 *csat* 方差的 20%。

如果要得到某一回归的标准化回归系数（"beta 权数"），加上选项 **beta** 即可。标准化回归系数是我们在一个所有变量都被转化为标准分（平均数为 0，标准差等于 1）后的回归中所看到的系数。

```
. regress csat expense percent income high college, beta

      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  5,    45) =   42.23
       Model |  184663.309     5  36932.6617           Prob > F      =  0.0000
    Residual |  39351.2012    45  874.471137           R-squared     =  0.8243
-------------+------------------------------           Adj R-squared =  0.8048
       Total |   224014.51    50   4480.2902           Root MSE      =  29.571

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|                     Beta
-------------+----------------------------------------------------------------
     expense |   .0033528   .0044709     0.75   0.457                  .070185
     percent |  -2.618177   .2538491   -10.31   0.000                -1.024538
      income |   .0001056   .0011661     0.09   0.928                 .0101321
        high |   1.630841    .992247     1.64   0.107                 .1361672
     college |   2.030894   1.660118     1.22   0.228                 .1263952
       _cons |   851.5649   59.29228    14.36   0.000                        .
------------------------------------------------------------------------------
```

标准化的回归方程为

预测值 $csat^*$ = 0.07$expense^*$ − 1.024 5$percent^*$ + 0.01$income^*$ +

$0.136high^{*} + 0.126college^{*}$

这里，*csat* ∗、*expense* ∗ 等为这些变量的标准分形式。我们可以按以下方式来解释 *percent* 的标准化系数：

$b_2 * = -1.0245$：在 *expense*、*income*、*high* 和 *college* 不变条件下，参加 SAT 考试的高中毕业生比例(*percent*)的每一标准差单位的提高将导致 SAT 平均分数预测值上 1.024 5 个标准差单位的下降。

这个回归在 F 和 t 检验值、R^2 和其他方面其实都保持不变。

预测值及残差

在任一回归分析之后，命令 **predict** 可以获得预测值、残差和其他对案例的统计量。假设我们已经对综合 SAT 成绩就其影响最强的单个解释变量进行了回归：

```
. regress csat percent
```

现在，创建一个叫 *yhat* 的新变量来存放从回归中得到预测的 *y* 值。

```
. predict yhat
. label variable yhat "Predicted mean SAT score"
```

通过 **resid** 选项，我们还可以创建另一个新变量来存放残差，这里取名为 e：

```
. predict e, resid
. label variable e "Residual"
```

我们也可以改用两条 **generate** 命令获得同样的 *y* 预测值以及残差。

```
. generate yhat0 = _b[_cons] + _b[percent]*percent
. generate e0 = csat - yhat0
```

Stata 会暂时“记住”回归系数和最近回归产生的其他信息。因此，_b[*varname*] 就是指自变量 *varname* 的回归系数。_b[_con] 则是指常数项系数(通常，就是指 y 的截距)。这些暂存值在编程和一些高级应用中十分有用，但作为最重要的目的，**predict** 免去了我们用“手工”方式创建 *yhat0* 和 *e0*。

残差包含了模型在哪里拟合较差的信息，因而对诊断分析或者排除故障的分析很重要。这样的分析可以从对残差的排序和检查开始。当我们高估了观察值就会出现负的残差。也就是说，当我们基于参加测试的学生比例去预测时，有些州的实际平均综合 SAT 成绩就会低于我们的预测值。为了列出残差最低的五个州，键入命令：

```
. sort e
. list state percent csat yhat e in 1/5
```

	state	percent	csat	yhat	e
1.	South Carolina	58	832	894.3333	-62.3333
2.	West Virginia	17	926	986.0953	-60.09526
3.	North Carolina	57	844	896.5714	-52.5714
4.	Texas	44	874	925.6666	-51.66666
5.	Nevada	25	919	968.1905	-49.19049

残差最低的四个州都属于南方，表明我们可以在一定程度上通过考虑地区因素来改进我

们的模型,更好地理解平均 SAT 成绩的差异。

当实际的 *y* 值高于预测值的时候会出现正的残差。因为数据已经按 *e* 排序,为了列出最高的五个残差,我们加上选择条件

in -5/l

选择条件中" **-5**"表示倒数第 5 个观察值,英文字母"el"(注意这里不是数字"1")表示最后一个(last)观察值。选择条件 **in 47/l** 或者 **in 47/51** 可以完成同样的事情。

```
. list state percent csat yhat e in -5/l

     +--------------------------------------------------------+
     |         state   percent   csat       yhat          e  |
     |--------------------------------------------------------|
 47. | Massachusetts        79    896   847.3333   48.66673  |
 48. |   Connecticut        81    897   842.8571   54.14292  |
 49. |  North Dakota         6   1073   1010.714   62.28567  |
 50. | New Hampshire        75    921   856.2856   64.71434  |
 51. |          Iowa         5   1093   1012.952   80.04758  |
     +--------------------------------------------------------+
```

predict 也可以从最近拟合的模型中导出其他统计量。下面是一些可以在 **anova** 或者 **regress** 后使用的 **predict** 选项。

. predict *new*	预测 *y* 值。**predict *new*, xb** 意味着做同样的事情(**xb** 表示 X 向量乘以 b,其实就是 *y* 预测值向量)。
. predict *new*, cooksd	Cook 的 *D* 影响统计量。
. predict *new*, covratio	*COVRATIO* 影响统计量,指每个观测值对估计值的方差协方差矩阵的影响。
. predict *DFx1* , dfbeta(*x1*)	*DFBETA* 统计量测量每个观测值对自变量 *x1* 的系数的影响。
. predict *new*, dfits	*DFITS* 影响统计量。
. predict *new*, hat	测量杠杆作用(leverage)帽子矩阵(hat matrix)的对角线元素。
. predict *new*, resid	残差。
. predict *new*, rstandard	标准化残差。
. predict *new*, rstudent	学生分布(studentized)标准化(刀切法)残差。
. predict *new*, stdf	单个预测值 *y* 的标准误,有时候叫做预报标准误或者预测标准误。
. predict *new*, stdp	预测的 *y* 平均数的标准误。
. predict *new*, stdr	残差的标准误。
. predict *new*, welsch	Welsch 氏距离(Welsch's distance)影响统计量。

更多的选项还包括预测概率和期望值;键入 **help regress** 可以得到清单。所有的 **predict** 选项都创建案例统计量,并形成新变量(就像预测值和残差),并且样本中的每个观察案例都有这些变量的取值。

使用 **predict** 命令时,用户可以用一个自己选择的新变量名代替上述命令中的变量 *new*。例如,要获得 Cook 的 *D* 影响统计量值,键入:

```
. predict D, cooksd
```

或者通过键入如下命令,可以获得帽子矩阵的对角线元素:

```
. predict h, hat
```

`predict` 所创建变量的名称(例如,*yhat*,*e*,*D*,*h*)是任意的,并且由用户指定。正如 Stata 命令的其他元素,我们可以用最小数目的字母来缩写这些选项来使其唯一性地辨认。例如,

```
. predict e, resid
```

可以被简写为:

```
. pre e, re
```

回归的基本图形

这一节介绍一些基本图形,用户可以用它们来描绘一个回归模型或者检查其拟合的情况。第7章描述更多专门的图形,它们主要用来辅助回归后的诊断工作。

在简单回归中,预测值取决于回归方程定义的直线。通过按坐标描点并连接预测值,我们可以让这条直线显示出来。命令 `lfit`(线性拟合)自动地画出简单回归线。

```
. graph twoway lfit csat percent
```

通常,像图6.1那样在回归直线图上叠并一张散点图会更好地引起人们的注意。

```
. graph twoway lfit csat percent
      || scatter csat percent
      || , ytitle("Mean composite SAT score") legend(off)
```

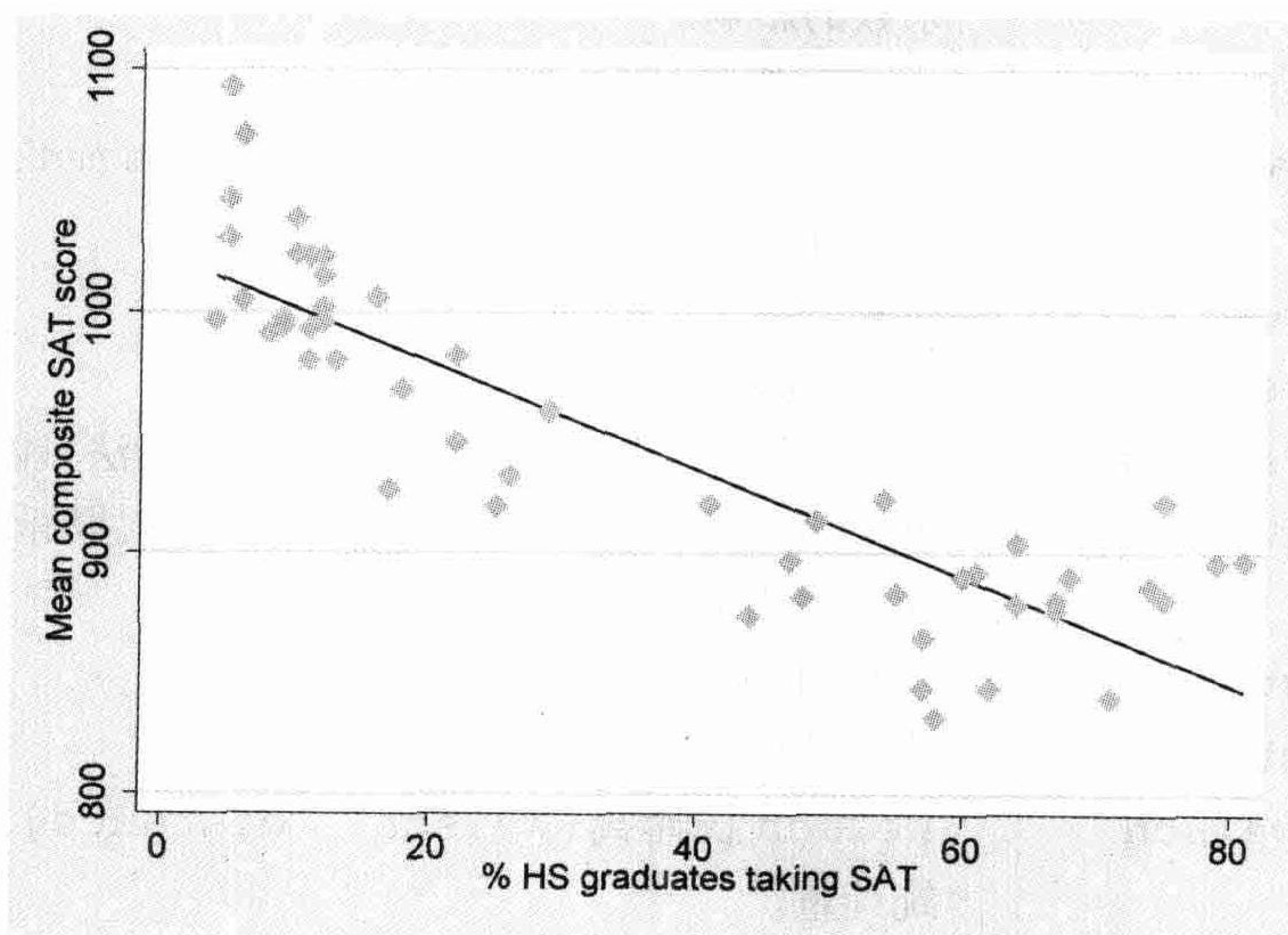


图6.1

我们可以用"手工"的办法画出图6.1,用回归后生成的预测值(*yhat*)和如下的命令形式即可:

```
. graph twoway mspline yhat percent, bands(50)
      ||  scatter csat percent
      ||  , legend(off) ytitle("Mean composite SAT score")
```

第二种方法需要做更多工作，但它为一些高级应用提供了更大的灵活性，比如说条件效应和非线性回归的标绘图。这种方法直接和预测值打交道，使得分析人员和数据以及回归模型正在做的事情保持更紧密的联系。当应用于线性预测值的时候，**graph twoway mspline**（拟合 50 个交叉中位数的立方样条曲线）将简单地画出一条直线，但当应用于非线性预测值的时候，也能同样很好地画出一条平滑的曲线来。

残差对预测值标绘图提供了有用的诊断工具（图 6.2）。在任意回归分析之后（还有其他模型，如方差分析），我们可以自动画出残差对拟合值（预测值）标绘图，只需键入：

```
. rvfplot, yline(0)
```

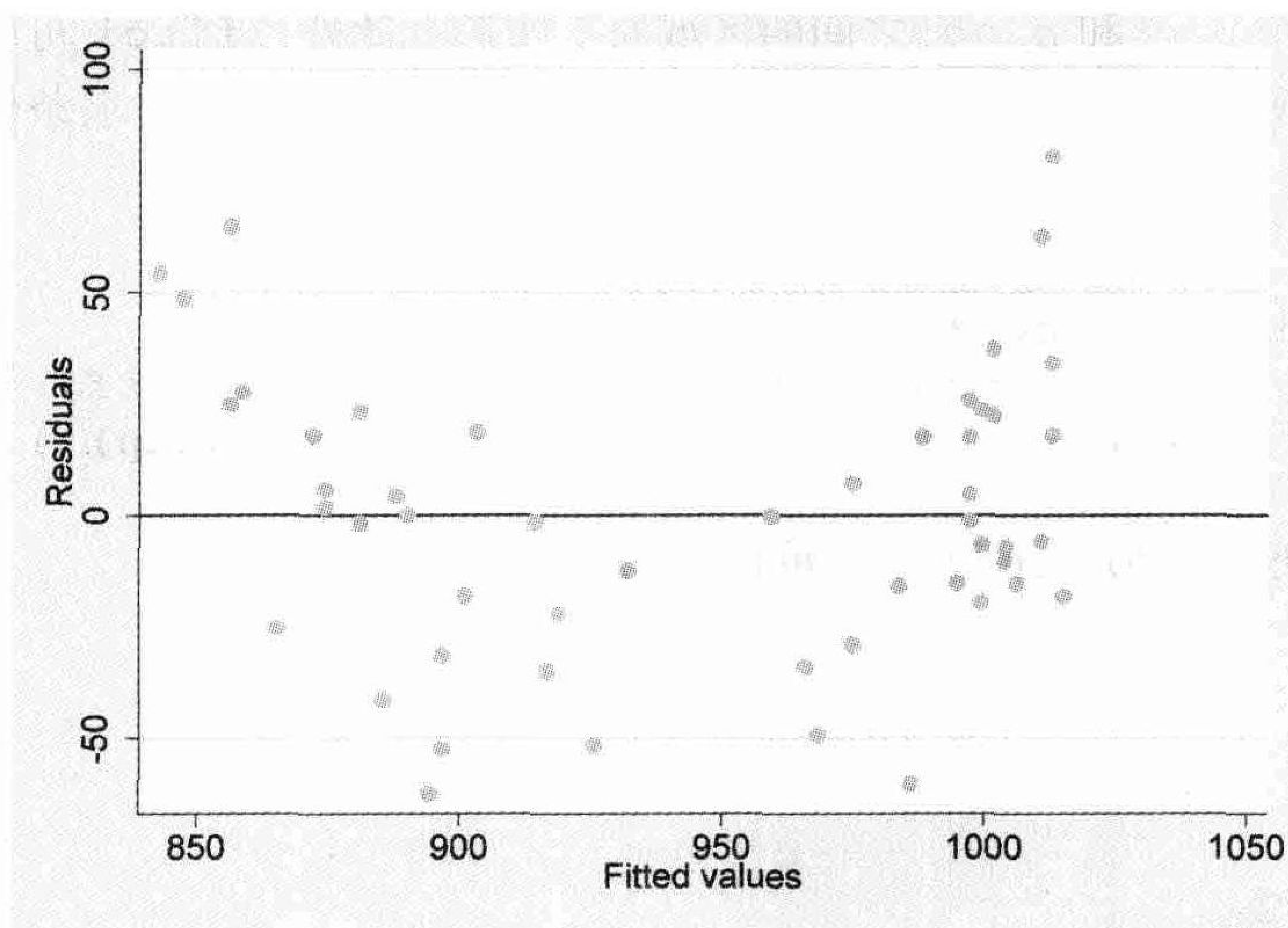


图 6.2

画出图 6.2 的“手工”办法是：

```
. graph twoway scatter e yhat, yline(0)
```

图 6.2 显示出我们目前的模型从整体上给出了数据的明显分布模式。在一开始，残差说预测误差看起来大部分是正值（由于预测过高），接下来大部分是负值，然后又是大部分为正值。后面的各节将要寻求一个更好地拟合数据的模型。

predict 可以为预测值创建两种标准误，它们有两种不同的应用。有时候可以通过“置信区间”和“预测区间”这样的名字来区别这些应用。在这样的区分背景下，置信区间表达了我们对估计 y 在给定 x 值（或者说在多元回归中给定 x 值的组合）下的条件平均数的不确定性。这种用途的标准误通过如下语句获得：

```
. predict SE, stdp
```

需要选择一个恰当的 t 值。由于自由度为 49，为了得到 95% 的置信区间，我们应该使用 $t = 2.01$，这可以查询 t 分布表或者直接询问 Stata：

```
. display invttail(49,.05/2)
2.0095752
```

于是，可以用以下命令计算出置信区间的下限：

```
. generate low1 = yhat - 2.01*SE
```

而置信区间上限为：

```
. generate high1 = yhat + 2.01*SE
```

简单回归中的置信区间带形状象沙漏，在 x 的平均数处最窄。我们可以像下面这样叠并多个 twoway 作图：

```
. graph twoway mspline low1 percent, clpattern(dash) bands(50)
      || mspline high1 percent, clpattern(dash) bands(50)
      || mspline yhat percent, clpattern(solid) bands(50)
      || scatter csat percent
      || , legend(off) ytitle("Mean composite SAT score")
```

阴影区域图(shaded-area range plots)(参见 **help twoway_rarea**)提供了不同的办法来画这样的图，它将 *low1* 和 *high1* 之间的区域涂上阴影。此外，**lfitci** 可以自动完成这个任务，并且完成区间带的计算，参见图 6.3。注意 **stdp** 选项，它要求条件平均数的置信区间带(实际上是默认值)。

```
. graph twoway lfitci csat percent, stdp
      || scatter csat percent, msymbol(O)
      || , ytitle("Mean composite SAT score")  legend(off)
      title("Confidence bands for conditional means (stdp)")
```

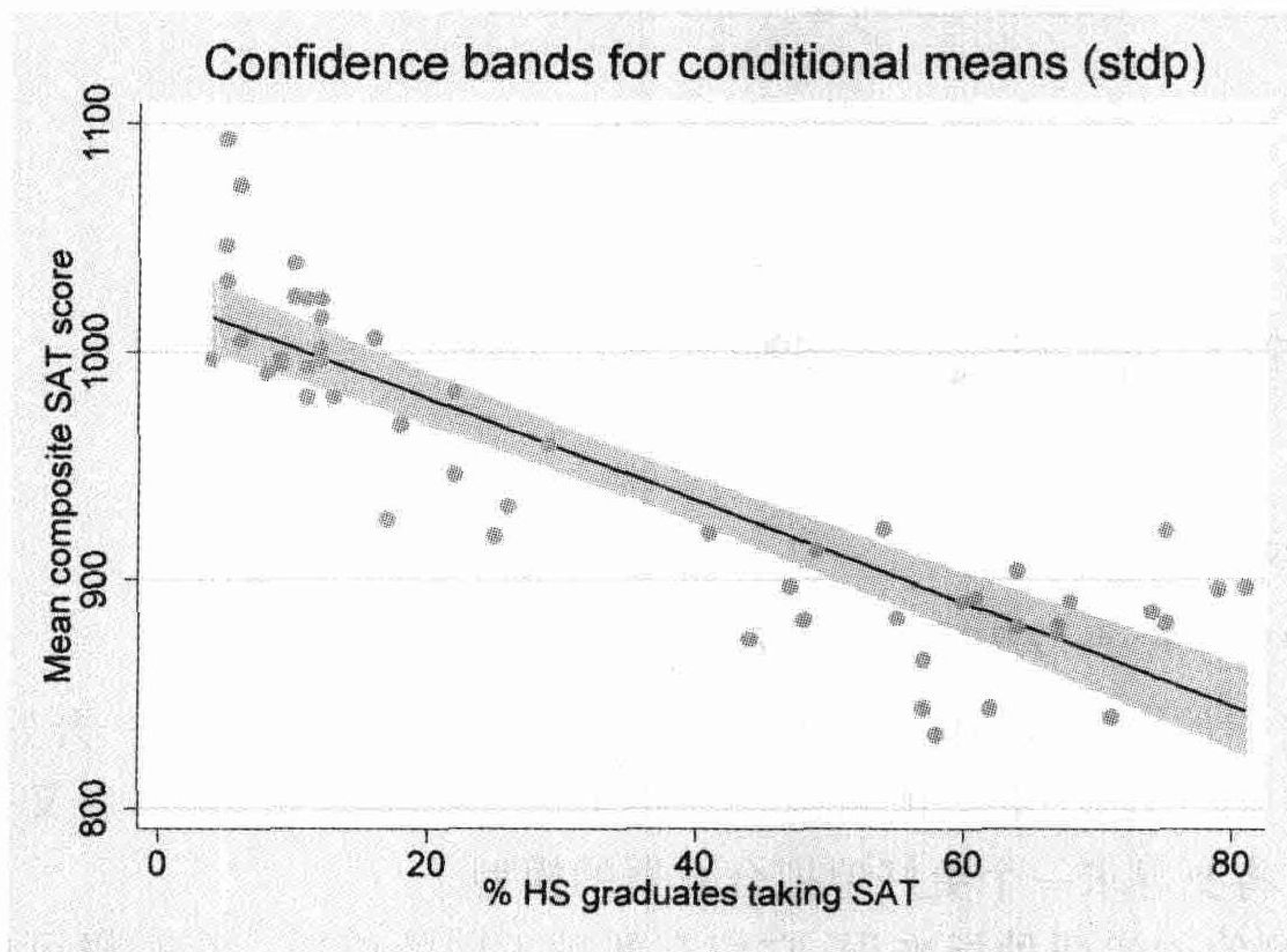


图 6.3

第二种类型的回归预测置信区间有时候被称为“预测区间”。它代表的是我们对用已知 x 值去估计未知 y 值的不确定性。这种情况下的标准误通过键入如下命令获得：

```
. predict SEyhat, stdf
```

图 6.4 使用带有 **stdf** 选项的 **lfitci** 命令画出这种预测区间带。像图 6.4 那样预测 y 的个体观测值本质上涉及更大的不确定性，因而比估计 y 的条件平均数(图 6.3)来说，其形成的置信区间带更宽。在两个例子中，最窄的波段都位于 x 的平均数处。

```
. graph twoway lfitci csat percent, stdf
      || scatter csat percent, msymbol(O)
      || , ytitle("Mean composite SAT score")  legend(off)
      title("Confidence bands for individual-case predictions (stdf)")
```

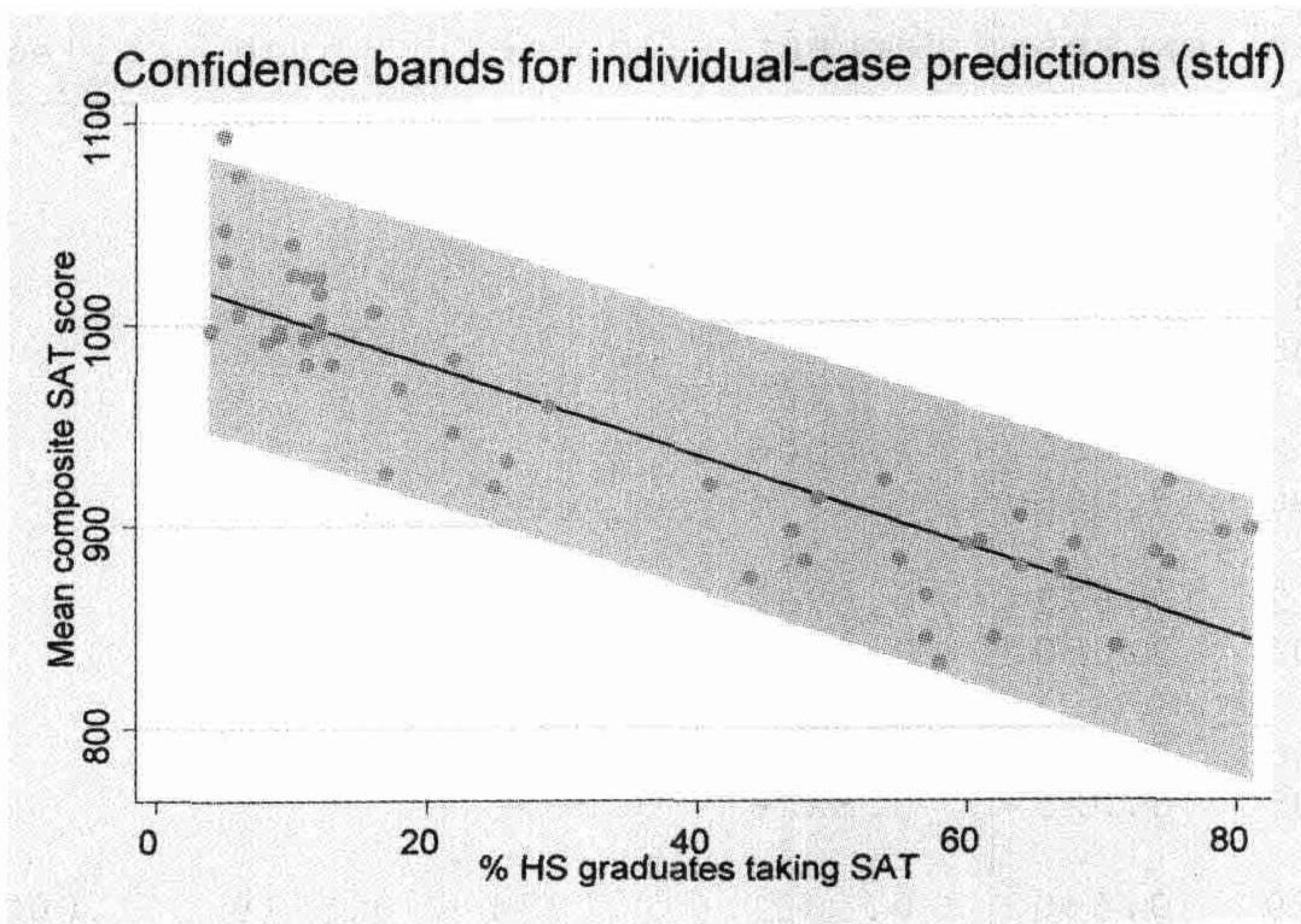


图 6.4

就像常规最小二乘回归中存在着其他置信区间和虚无假设检验一样，刚才描述的标准误和置信区间也有赖于关于独立同分布误差的假定(assumption of independent and identically distributed errors)。图 6.2 已经对这一假定提出了质疑，因此图 6.3 和图 6.4 的结果或许会形成误导。

相 关

命令 **correlate** 用来获得皮尔逊积距相关(Pearson product moment correlation)。

```
. correlate csat expense percent income high college

(obs=51)

             |     csat  expense  percent   income     high  college
-------------+------------------------------------------------------
        csat |   1.0000
     expense |  -0.4663   1.0000
     percent |  -0.8758   0.6509   1.0000
      income |  -0.4713   0.6784   0.6733   1.0000
        high |   0.0858   0.3133   0.1413   0.5099   1.0000
     college |  -0.3729   0.6400   0.6091   0.7234   0.5319   1.0000
```

correlate 仅仅使用在命令中列出的所有变量上都没有缺失值的那部分数据子集(对于这些特定变量，这一点不重要，因为它们都没有缺失值)。在这方面，**correlate** 命令和 **regress** 比较相似，如果给定同样的的变量清单，它们将使用同样的数据子集。虽然有些分析人员不使用回归和其他多变量分析技术，但他们可能想基于所有可获得的观察案例成对变量值来计算相关系数。命令 **pwcorr**(配对相关，pairwise correlation)可以完成这一任务，并提供 t 检验概率来检验每一项相关等于零的虚无假设。

```
. pwcorr csat expense percent income high college, sig
```

```
             |     csat  expense  percent   income     high  college
-------------+------------------------------------------------------
        csat |   1.0000
             |
             |
     expense |  -0.4663   1.0000
             |   0.0006
             |
     percent |  -0.8758   0.6509   1.0000
             |   0.0000   0.0000
             |
      income |  -0.4713   0.6784   0.6733   1.0000
             |   0.0005   0.0000   0.0000
             |
        high |   0.0858   0.3133   0.1413   0.5099   1.0000
             |   0.5495   0.0252   0.3226   0.0001
             |
     college |  -0.3729   0.6400   0.6091   0.7234   0.5319   1.0000
             |   0.0070   0.0000   0.0000   0.0000   0.0001
             |
```

这里我们值得去回忆一下，那就是如果我们从所有变量之间确实为 0 相关的总体中抽取许多随机样本，约有 5% 的样本相关仍将会在 0.05 水平"统计性显著"。有的分析人员去查看许多单个相关的检验，比如，列在 **pwcorr** 结果矩阵中的那些系数，因为确认有一小部分系数在 0.05 水平上显著，由此犯第一类错误的风险要比 0.05 高得多。这一问题称为"多重比较谬误"(multiple comparison fallacy)。**pwcorr** 命令提供的 Bonferroni 检验和 Šidák 检验这两种方法将多重比较纳入考虑来调整显著性水平。其中，Šidák 方法更为精确。

```
. pwcorr csat expense percent income high college, sidak sig
```

```
             |     csat  expense  percent   income     high  college
-------------+------------------------------------------------------
        csat |   1.0000
             |
             |
     expense |  -0.4663   1.0000
             |   0.0084
             |
     percent |  -0.8758   0.6509   1.0000
             |   0.0000   0.0000
             |
      income |  -0.4713   0.6784   0.6733   1.0000
             |   0.0072   0.0000   0.0000
             |
        high |   0.0858   0.3133   0.1413   0.5099   1.0000
             |   1.0000   0.3180   0.9971   0.0020
             |
     college |  -0.3729   0.6400   0.6091   0.7234   0.5319   1.0000
             |   0.1004   0.0000   0.0000   0.0000   0.0009
```

将上表中的检验概率和前面 **pwcorr** 结果中的概率相比较，可以知道出现了多大程度的调整。通常来说，我们纳入相关分析的变量越多，调整后的概率超过原来概率就越多。相关公式参见《基础参考手册》有关 **oneway** 的讨论。

correlate 自身提供了几个重要的选项。加上 **covariance** 选项将产生一个方差协方差矩阵来代替相关矩阵。

```
. correlate w x y z, covariance
```

在回归分析后键入下面的命令将显示估计系数之间的相关矩阵,这有时可用于诊断多元共线性(见第 7 章):

```
. correlate, _coef
```

下列命令将显示估计系数的方差协方差矩阵,标准误便可以由此导出:

```
. correlate, _coef covariance
```

皮尔逊相关系数测量一条 OLS 回归线对数据的拟合优度。因此,这些系数具有和 OLS 同样的假定和弱点,并且和 OLS 一样,在没有检查相应的散点图之前不宜进行解释。散点图矩阵提供了完成这一任务的快捷方法,它使用和相关矩阵同样的结构。图 6.5 显示了对应先前 **pwcorr** 矩阵的散点图矩阵。这里仅画出了矩阵的下三角部分,并且用加号作为散点标志。这里,我们去掉了 y 轴和 x 轴的标签以保持图形的整洁。

```
. graph matrix csat expense percent income high college,
      half msymbol(+) maxis(ylabel(none) xlabel(none))
```

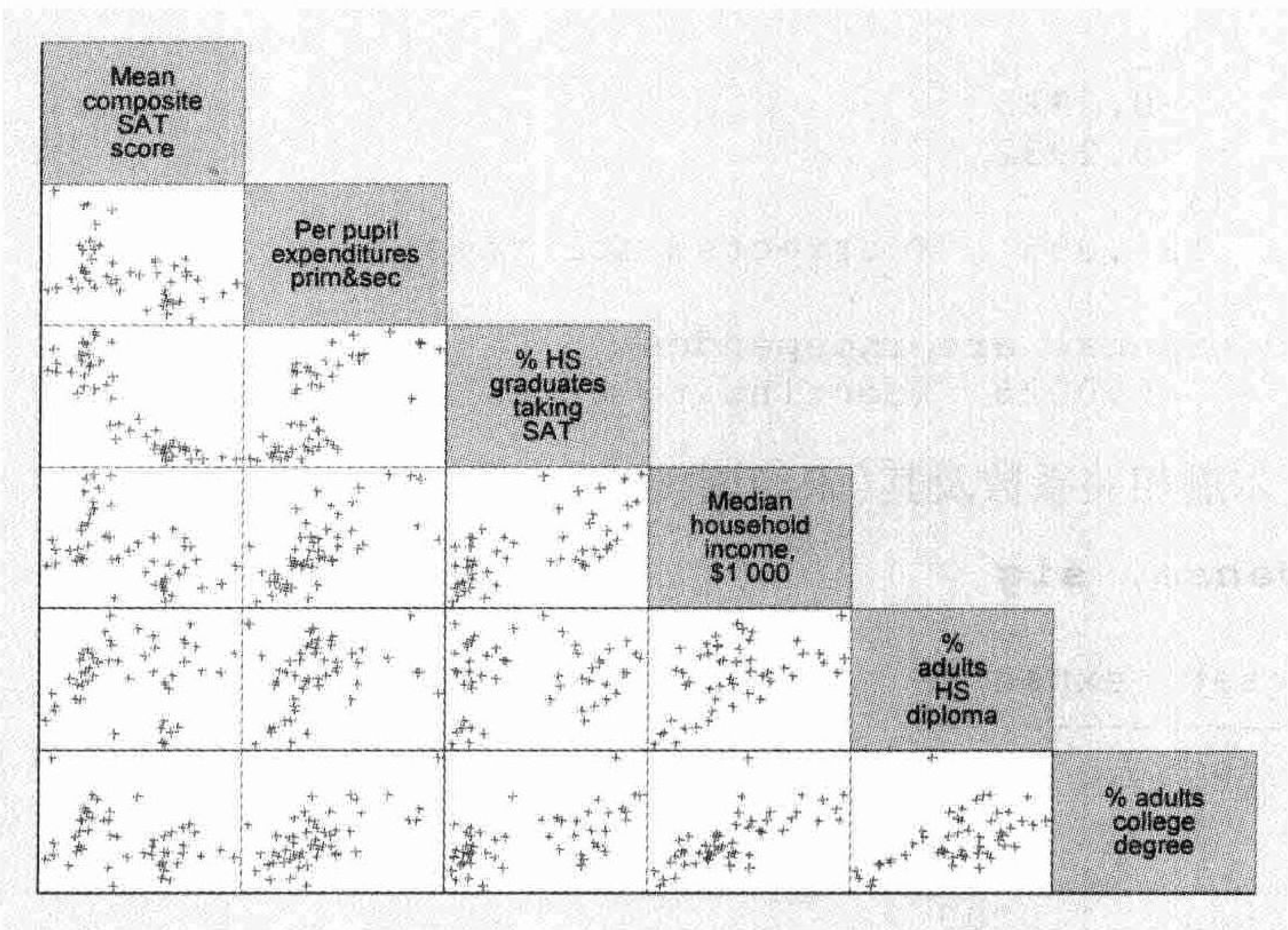


图 6.5

我们需要对命令进行条件限制来获得与 **correlate** 相关矩阵对应的散点图矩阵,因为所有带有缺失值的观察要被事先删除。如果所有的变量都有缺失值,我们应该键入这样一条命令:

```
. graph matrix csat expense percent income high college if
     csat <. & expense <. & income <.& high<. & college <.
```

为了减少混淆和错误的可能性,很值得创建一个新数据集并仅仅保存那些没有缺失值的观察数据:

```
. keep if csat < . & expense < . & income < . & high < .
     & college < .
. save nmvstate
```

在这个示例中,我们直接用新名字的文件来保存简化后的数据,以避免不小心覆盖和丢失旧文件中更为完整的数据。另一个可以选择的办法是使用 **drop** 代替 **keep** 来删除缺失值:

```
. drop if csat >= . | expense >= . | income >= . | high >= .
     | college >= .
. save nmvstate
```

除了皮尔逊相关系数，Stata 还可以计算几种基于序次(rank-based)的等级相关系数。它们可以用来测量定序变量之间的联系，或者作为皮尔逊相关的一种抗特异值的稳健指标。把测量变量转换成序次后，*csat* 和 *expense* 之间的斯比尔曼等级相关等价于皮尔逊相关。键入：

```
. spearman csat expense

 Number of obs =       51
Spearman's rho =       -0.4282

Test of Ho: csat and expense are independent
    Prob > |t| =        0.0017
```

Kendall 的 τ_a(读 tau-a)、τ_b(读 tau-b)的等级相关也可以从这些数据中容易地得到，尽管它们的计算在数据集较大时变得很慢。

```
. ktau csat expense

  Number of obs =       51
Kendall's tau-a =       -0.2925
Kendall's tau-b =       -0.2932
Kendall's score =     -373
    SE of score =      123.095    (corrected for ties)

Test of Ho: csat and expense are independent
     Prob > |z| =        0.0025   (continuity corrected)
```

作为对比，这里提供了皮尔逊相关系数及其(未经调整的)*P* 值：

```
. pwcorr csat expense, sig

             |     csat  expense
-------------+------------------
        csat |   1.0000
             |
             |
     expense |  -0.4663   1.0000
             |   0.0006
             |
```

在这一示例中，**spearman** 相关值(-0.428 2)和 **pwcorr** 相关值(-0.466 3)都比 **ktau** 相关值(-0.292 5 或-0.293 2)更大。这三者在拒绝没有关联这一虚无假设上的结论是一致的。

假设检验

在 **regress** 输出表中有两种类型的假设检验。与其他通常的假设检验一样，它们都是基于这样一个假定，即所分析的样本观察案例是从一个无限大的总体中随机并且独立地抽取出来的。

1.整体 *F* 检验：回归表右上部的 *F* 统计量评价这样的虚无假设，即在总体中，模型中所有 *x* 变量的回归系数都等于 0。

2.单个 *t* 检验：回归表的第三栏和第四栏包含了单个回归系数的 *t* 检验。它们评估的虚无假设是：在总体中，每一个特定的 *x* 变量的系数等于 0。

t 检验概率是双侧概率。对单侧概率，把 *P* 值除以 2 即可。

除了这些标准的 *F* 检验和 *t* 检验，Stata 也可以对用户指定的假设进行 *F* 检验。

test 命令引用最近执行的模型拟合命令的结果,比如说 **anova** 或 **regress**。举一个例子,下述回归报告中的单个 t 检验指出,成人中具有高中以上文凭的比例(*high*)和具有大学文凭的比例(*college*)都对综合 SAT 成绩没有显著的单独影响。

```
. regress csat expense percent income high college
```

但是,从理论上看,两个因素都反映州人口的的教育水平,并且出于某种目的,我们可能想检验是不是两者的影响同时为零。为了实现这一点,我们在一开始"悄悄地"(加上了 **quietly** 选项)重做了多元回归,因为我们并不需要再次看到全部的输出结果。然后,我们使用 **test** 命令:

```
. quietly regress csat expense percent income high college
. test high college

(1)   high = 0.0
(2)   college = 0.0

       F(  2,     45) =     3.32
            Prob > F =     0.0451
```

和单个虚无假设不同,可以有根据地拒绝关于 *high* 和 *college* 同时等于零的这个联合虚无假设。当我们有几个概念上相关的解释变量,或者由于多元共线性(第 7 章)导致单个系数估计显示出不可靠时,这种关于系数子集的检验就非常有用了。

test 还能够复制整体的 F 检验:

```
. test expense percent income high college
```

test 还能复制出单个系数检验:

```
. test expense
. test percent
. test income
```

,等等。**test** 在高级任务中更加有用,这些高级应用包括:

1.检验某一系数是否等于指定的常数。例如,检验 *income* 的系数等于 1 这一虚无假设($H_0: \beta_3 = 1$),以代替通常情况下关于其等于 0 的虚无假设($H_0: \beta_3 = 0$),键入:

```
. test income = 1
```

2.检验两个系数是否相等。例如,下面的命令评价虚无假设($H_0: \beta_4 = \beta_5$):

```
. test high = college
```

3.最后,**test** 还能接受某些代数表达式。我们可以要求检验一个虚无假设($H_0: \beta_3 = (\beta_4 + \beta_5)/100$):

```
. test income = (high + college)/100
```

更多的信息和示例请参考 **help test**。

虚拟变量

分类变量被表示成一个或者多个"虚拟变量"(dummy variable),它们都是取值为{0,1}之一的二分变量。这时,它们可以成为回归中的预测变量。例如,我们有理由猜想在州平均 SAT 成绩中存在着地区差异。如果我们加上 **gen**(创建,**genarate** 缩

写)选项,命令 **tabulate** 将为列表中分类变量的每一类别创建一个虚拟变量。下面,我们根据四种分类的变量 *region* 来创建四个虚拟变量。这些虚拟变量取名为 *reg1*、*reg2*、*reg3* 和 *reg4*。对西部的州来说,虚拟变量 *reg1* 等于 1,而其他的州则等于 0;对东北部的州来说,*reg2* 等于 1,其余的州则等于 0;如此等等。

```
. tabulate region, gen(reg)

Geographica |
   l region |      Freq.     Percent        Cum.
------------+-----------------------------------
       West |         13       26.00       26.00
   N. East  |          9       18.00       44.00
      South |         16       32.00       76.00
    Midwest |         12       24.00      100.00
------------+-----------------------------------
      Total |         50      100.00

. describe reg1-reg4

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
reg1            byte   %8.0g                  region==West
reg2            byte   %8.0g                  region==N. East
reg3            byte   %8.0g                  region==South
reg4            byte   %8.0g                  region==Midwest
. tabulate reg1

region==Wes |
          t |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         37       74.00       74.00
          1 |         13       26.00      100.00
------------+-----------------------------------
      Total |         50      100.00

. tabulate reg2

 region==N. |
       East |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         41       82.00       82.00
          1 |          9       18.00      100.00
------------+-----------------------------------
      Total |         50      100.00
```

将 *csat* 对一个虚拟变量 *reg2*(东北部)进行回归等价于执行在 *reg2* 各类别上 *csat* 平均数是否相同的两样本 *t* 检验。也就是在问:东北部州的 *csat* 平均数是否和美国其他的州一样?

```
. regress csat reg2

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  1,    48) =    9.50
       Model |  35191.4017     1  35191.4017           Prob > F      =  0.0034
    Residual |  177769.978    48  3703.54121           R-squared     =  0.1652
-------------+------------------------------           Adj R-squared =  0.1479
       Total |   212961.38    49  4346.15061           Root MSE      =  60.857

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        reg2 |   -69.0542   22.40167    -3.08   0.003    -114.0958   -24.01262
       _cons |   958.6098   9.504224   100.86   0.000     939.5002    977.7193
------------------------------------------------------------------------------
```

虚拟变量回归系数的 t 统计量（$t=-3.08, P=0.003$）表明存在显著不同。根据这一回归，东北部各州的平均 SAT 成绩要低 69.054 2 分。我们从简单的 t 检验也能获得同样的结果（$t=-3.08, P=0.003$），它还显示出东北部各州的平均数为889.555 6，而所有其他州的平均数为 958.609 8，两者之间的差异为 69.054 2。

```
. ttest csat, by(reg2)

Two-sample t test with equal variances

------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |      41    958.6098    10.36563    66.37239      937.66    979.5595
       1 |       9    889.5556    4.652094    13.95628    878.8278    900.2833
---------+--------------------------------------------------------------------
combined |      50      946.18    9.323251    65.92534    927.4442    964.9158
---------+--------------------------------------------------------------------
    diff |             69.0542    22.40167                24.01262    114.0958
------------------------------------------------------------------------------
Degrees of freedom: 48

                 Ho: mean(0) - mean(1) = diff = 0

     Ha: diff < 0           Ha: diff != 0           Ha: diff > 0
       t =   3.0825           t =   3.0825           t =   3.0825
   P < t =   0.9983     P > |t| =   0.0034       P > t =   0.0017
```

然而，一旦我们控制了参加考试学生的比例这一变量时，这一结论就被证明是靠不住的。为此，我们将 *csat* 对 *reg2* 和 *percent* 进行多元回归：

```
. regress csat reg2 percent

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  2,    47) =  107.18
       Model |  174664.983     2  87332.4916           Prob > F      =  0.0000
    Residual |  38296.3969    47  814.816955           R-squared     =  0.8202
-------------+------------------------------           Adj R-squared =  0.8125
       Total |   212961.38    49  4346.15061           Root MSE      =  28.545

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        reg2 |   57.52437   14.28326     4.03   0.000     28.79016    86.25858
     percent |  -2.793009   .2134796   -13.08   0.000    -3.222475   -2.363544
       _cons |   1033.749   7.270285   142.19   0.000     1019.123    1048.374
------------------------------------------------------------------------------
```

现在，东北部地区变量 *reg2* 有了统计性非常显著的正值回归系数（$b=57.524\ 37$，$P<0.000\ 5$）。这说明早先取得的负系数是误导性的。虽然东北部各州的平均 SAT 值确实要低一些，但它们之所以低，是因为在东北部地区，较高比例的学生参加了这一测试。在许多其他地区的州，只有较少的、更加"精英化"的学生群体才参加这一测试，他们通常还不到高中高年级学生的 20% 。然而，在东北部各州大部分学生（64%~81%）都参加测试。一旦我们对参加测试比例的差异加以调整，东北部各州的 SAT 成绩实际上反而更高一些。

写出回归方程，并代入虚拟变量值，有助于理解虚拟变量回归结果。对东北部各州来说，回归方程为：

$$
\begin{aligned}
\text{预测值 } csat &= 1\ 033.7 + 57.5 reg2 - 2.8 percent \\
&= 1\ 033.7 + 57.5 \times 1 - 2.8 percent \\
&= 1\ 091.2 - 2.8 percent
\end{aligned}
$$

对其他的州，其预测的 *csat* 值在任一给定的 *percent* 水平上都将低 57.5 分：

预测值 $csat = 1\,033.7 + 57.5 \times 0 - 2.8percent$

$= 1\,033.7 - 2.8percent$

在这种模型中的虚拟变量被称为“截距虚拟变量”(intercept dummy variable),因为它们描述了 y 轴上截距或者说常数的变化。

根据一个具有 k 个类别的分类变量,我们可以定义 k 个虚拟变量,但它们中有一个虚拟变量是多余的。例如,一旦我们知道某个州在西部、东北部和中部这三个虚拟变量的取值,我们就已经猜到它在南部虚拟变量上的取值。正因为如此,只有 $k-1$ 个虚拟变量(在这个关于 *region* 的案例中是3个)可以被纳入回归。如果我们试图包括所有可能的虚拟变量,Stata 将会自动除去其中一个,否则将会因为多元共线性而导致计算无法进行。

```
. regress csat reg1 reg2 reg3 reg4 percent

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  4,    45) =   64.61
       Model |  181378.099     4  45344.5247           Prob > F      =  0.0000
    Residual |  31583.2811    45  701.850691           R-squared     =  0.8517
-------------+------------------------------           Adj R-squared =  0.8385
       Total |   212961.38    49  4346.15061           Root MSE      =  26.492

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        reg1 |  -23.77315   11.12578    -2.14   0.038    -46.18162   -1.364676
        reg2 |   25.79985   16.96365     1.52   0.135    -8.366693    59.96639
        reg3 |  -33.29951   10.85443    -3.07   0.004    -55.16146   -11.43757
        reg4 |  (dropped)
     percent |  -2.546058   .2140196   -11.90   0.000    -2.977116   -2.115001
       _cons |   1047.638   8.273625   126.62   0.000     1030.974    1064.302
------------------------------------------------------------------------------
```

不管我们(或者是 Stata)选择省略哪一个虚拟变量,模型的拟合情况(包括 R^2、F 检验、预测以及残差)实质上保持不变。在这个示例中,中西部虚拟变量(*reg4*)被省略了。*reg1*,*reg2* 以及 *reg3* 的回归系数告诉我们,在任一给定的 *percent* 水平上,预测的平均 SAT 成绩大致如下:

西部(*reg1* = 1)比中西部低 23.8 分;

东北部(*reg2* = 1)比中西部高 25.8 分;以及

南部(*reg3* = 1)比中西部低 33.3 分。

西部和南部都显著地低于中西部,但东北部并不是这样。

另一个命令 **areg** 拟合同样的模型,但它并不需要先创建虚拟变量后再进行。它能“吸收”一个 k 分类变量的效应,例如,*region*。其模型拟合、对吸收变量(absorbed variable)的 F 检验和其他关键方面都和我们通过明确设立虚拟变量获得的结果完全一样。但是,要注意的是,**areg** 并不提供单个虚拟变量的系数估计。

```
. areg csat percent, absorb(region)
                                                       Number of obs =      50
                                                       F(  1,    45) =  141.52
                                                       Prob > F      =  0.0000
                                                       R-squared     =  0.8517
                                                       Adj R-squared =  0.8385
                                                       Root MSE      =  26.492

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     percent |  -2.546058   .2140196   -11.90   0.000    -2.977116   -2.115001
       _cons |   1035.445    8.38689   123.46   0.000     1018.553    1052.337
-------------+----------------------------------------------------------------
      region |          F(3, 45) =      9.465   0.000          (4 categories)
```

虽然 **areg** 的输出信息比带有明确虚拟变量的回归要少,但它有两个优点。首先,它加快了探索进程,因为很快就可以对一个虚拟变量是否值得研究作出反馈。其次,当我们所关心的变量有许多取值,那么为每一种取值创建虚拟变量对特定 Stata 配置来说,可能会导致变量太多或者模型过大。于是,**areg** 可以按照通常的数据限制和矩阵规模来工作。

然而,明确的虚拟变量也有其他优点,比如,可以在模型中纳入交互效应。所谓"斜率虚拟变量"(slope dummy variables)的交互项可以通过将虚拟变量乘以测量型变量来形成。例如,为了把东北部相对其他区域这个地区因素和 *percent* 的交互相应纳入模型,我们创建了一个斜率虚拟变量,取名 *reg2perc*。

```
. generate reg2perc = reg2 * percent
(1 missing value generated)
```

新变量 *reg2perc* 对于东北部的州来说等于 *percent*,而对于其他州来说等于 0。我们可以把这个交互项加入回归解释变量表:

```
. regress csat reg2 percent reg2perc

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  3,    46) =   82.27
       Model |  179506.19      3  59835.3968           Prob > F      =  0.0000
    Residual |  33455.1897    46  727.286733           R-squared     =  0.8429
-------------+------------------------------           Adj R-squared =  0.8327
       Total |  212961.38     49  4346.15061           Root MSE      =  26.968

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        reg2 |  -241.3574   116.6278    -2.07   0.044     -476.117   -6.597821
     percent |  -2.858829   .2032947   -14.06   0.000     -3.26804   -2.449618
    reg2perc |   4.179666   1.620009     2.58   0.013     .9187559    7.440576
       _cons |   1035.519   6.902898   150.01   0.000     1021.624    1049.414
------------------------------------------------------------------------------
```

交互项统计性显著($t = 2.58, P = 0.013$)。由于这一分析同时包括截距虚拟变量(*reg2*)和斜率虚拟变量(*reg2perc*),值得我们写出方程。对东北部各州的回归方程大致为:

$$\text{预测值 } csat = 1\,035.5 - 241.4reg2 - 2.9percent + 4.2reg2perc$$
$$= 1\,035.5 - 241.4 \times 1 - 2.9percent + 4.2 \times 1 \times percent$$
$$= 794.1 + 1.3percent$$

对其他州则为:

$$\text{预测值 } csat = 1\,035.5 - 241.4 \times 0 - 2.9percent + 4.2 \times 0 \times percent$$
$$= 1\,035.5 - 2.9percent$$

交互项意味某一变量变化带来的效应将依赖于其他变量的取值。从这一回归来看,它显示了 *percent* 在东北部的州中具有一个较弱的正影响,然而对其他州而言,它的影响要强一些,并且是负向的。

为了将一个斜率和截距虚拟变量回归的结果可视化,我们有几种作图的可能性。甚至不用拟合模型,我们就可以像下面这样用 **lfit** 来完成这项工作,其结果可以在图 6.6 中看到。

```
. label define reg2 0 "other regions" 1 "Northeast"
. label values reg2 reg2
. graph twoway lfit csat percent
      ||  scatter csat percent
      ||  , by(reg2, legend(off) note(""))
     ytitle("Mean composite SAT score")
```

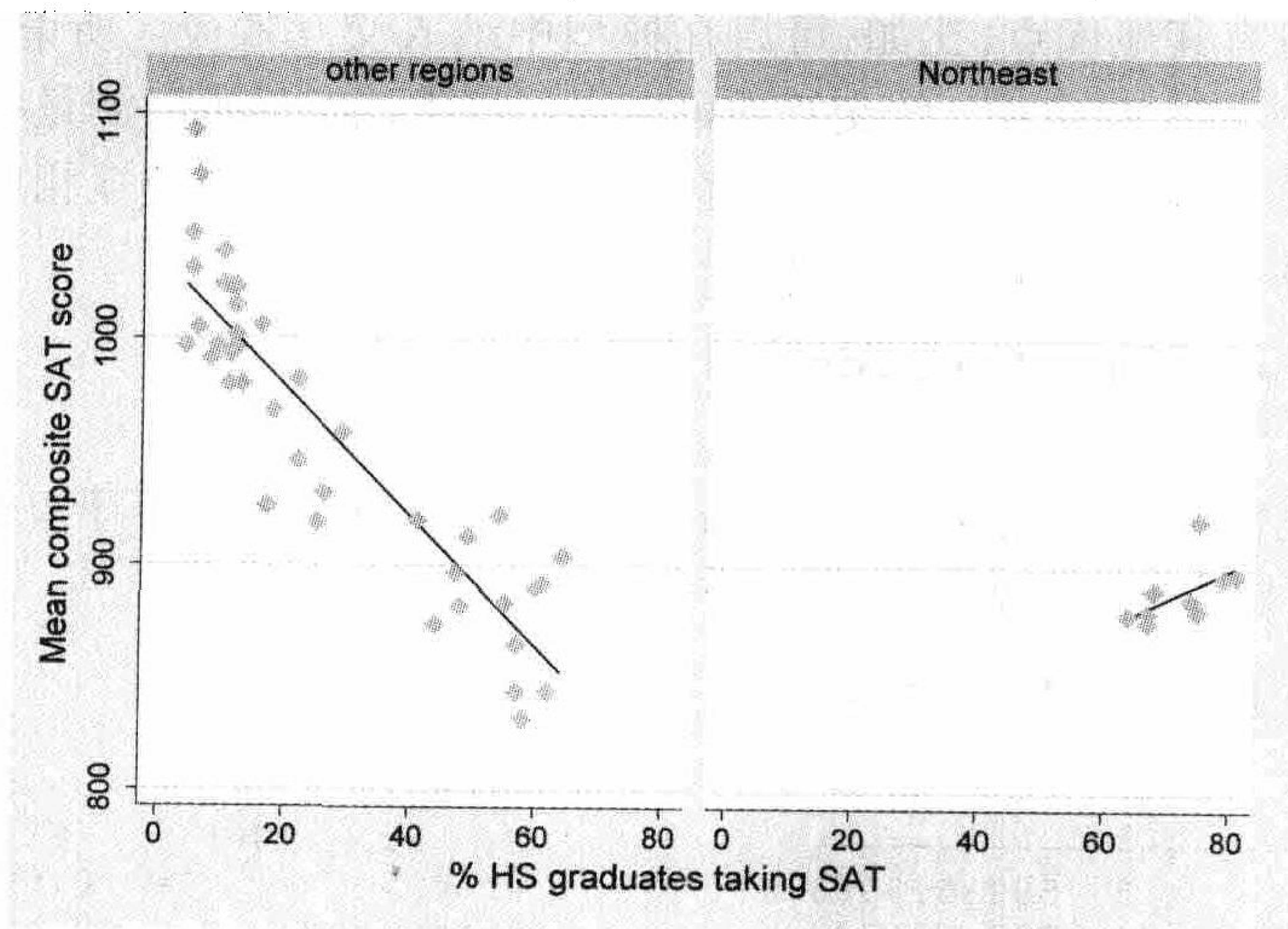


图 6.6

或者，我们可以先拟合模型，计算预测值，并使用它们来画出一张更为精致的图，就像图 6.7 那样。两条 **mspline** 命令中的 **bands(50)** 选项指定了基于 50 个垂直波段的中位样条，这些波段足够覆盖数据的范围。

```
. quietly regress csat reg2 percent reg2perc
. predict yhat1
. graph twoway scatter csat percent if reg2 == 0
     ||  mspline yhat1 percent if reg2 == 0, clpattern(solid)
     bands(50)
     ||  scatter csat percent if reg2 == 1, msymbol(Sh)
     ||  mspline yhat1 percent if reg2 == 1, clpattern(solid)
     bands(50)
     ||  , ytitle("Composite mean SAT score")
     legend(order(1 3) label(1 "other regions")
        label(3 "Northeast states") position(12) ring(0))
```

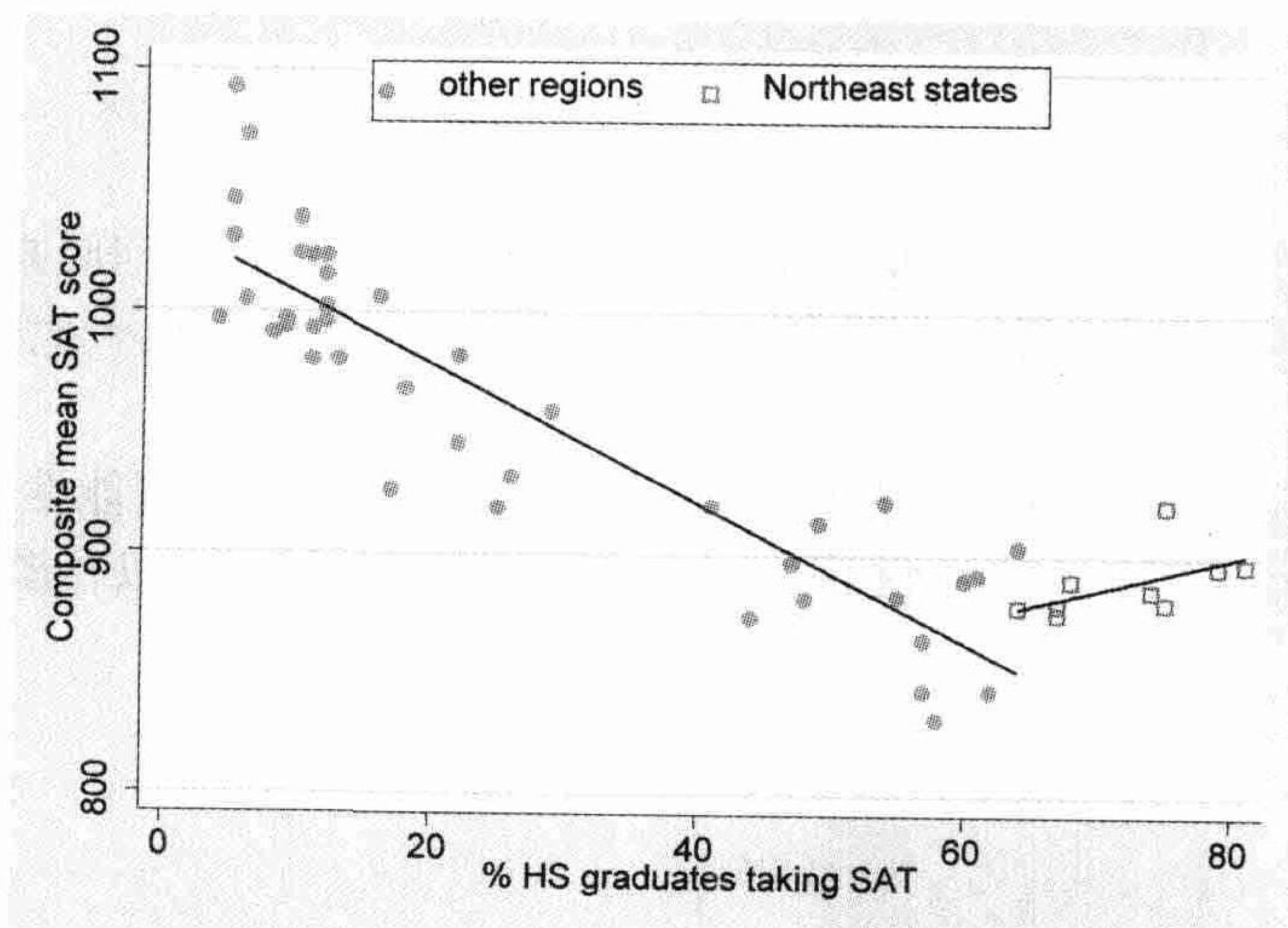


图 6.7

图 6.7 涉及 4 个重叠：两个散点图（东北部州和其他州分别的 *csat* 对 *percent* 图）和两个中位样条标绘图（连接预测值 *yhat1*，对东北部州以及其他州的 *percent* 作图）。选项 **msymbol(Sh)** 要求东北部的州标志为中空的四方形。**ytitle** 和 **legend** 选项简化了 *y* 轴标题和图例。如果按照它们的默认形式，二者都会显得拥挤不堪、看不清楚。

图 6.6 和图 6.7 都显示出在东北部的州和其他州之间存在显著差别，这些差别被我们的交互项捕获到了。这进一步引起了还存在哪些其他地区差异的问题。图 6.8 通过用不同的标志按 4 个地区画 *csat* 对 *percent* 的散点分布图探讨了这一问题。在这张图里面，中西部除了一个州（印第安那，Indiana）外其他都位于图的左边，并且看起来具有其独特的陡峭、负向的地域模式。南部的州的异质性最高。

```
. graph twoway scatter csat percent if reg1 ==1
      || scatter csat percent if reg2 ==1, msymbol(Sh)
      || scatter csat percent if reg3 == 1, msymbol(T)
      || scatter csat percent if reg4 == 1, msymbol(+)
      || , legend(position(1) ring(0) label(1 "West")
    label(2 "Northeast") label(3 "South") label(4 "Midwest"))
```

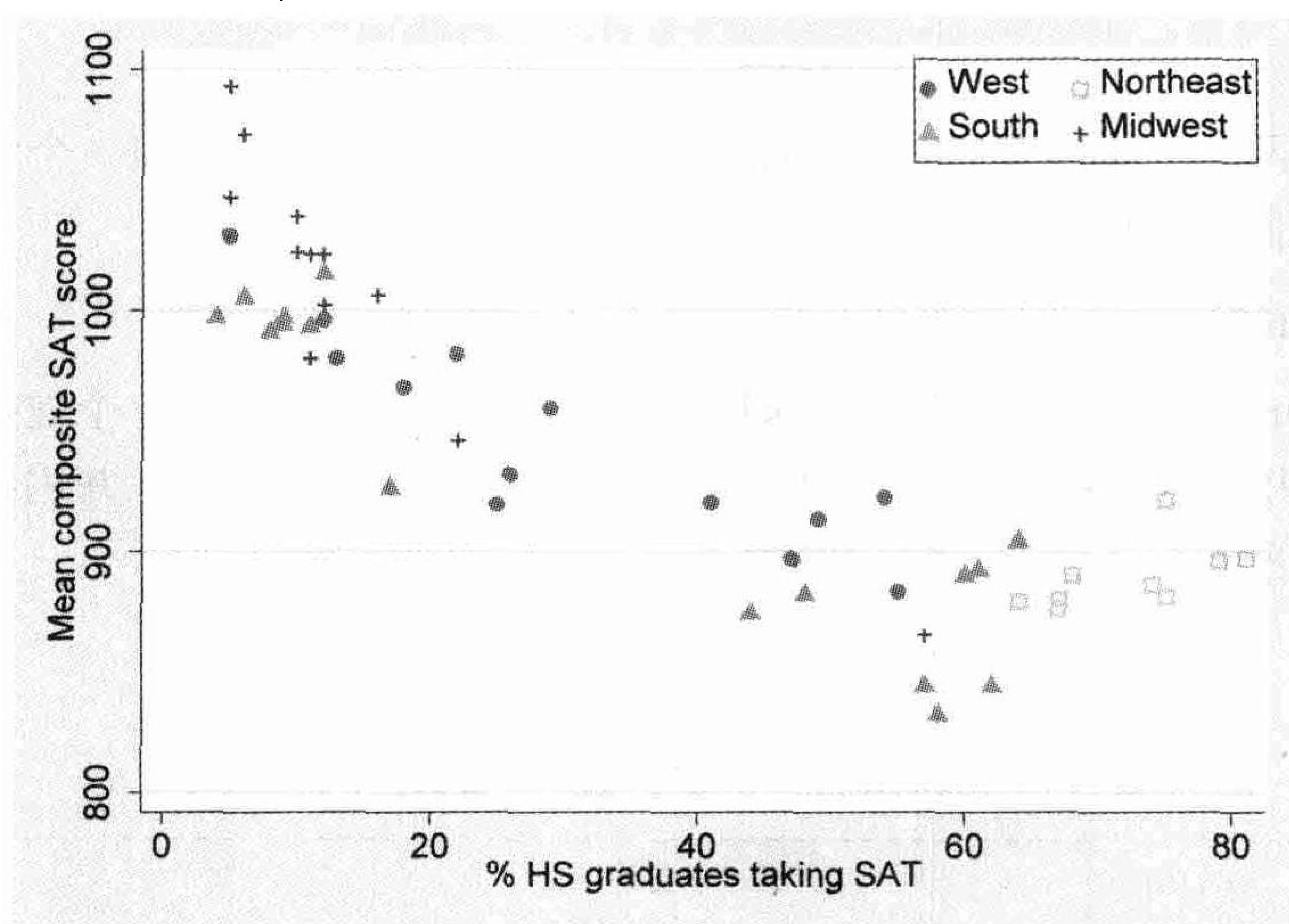


图 6.8

分类变量的自动标识和交互项

命令 **xi**（意为扩展交互项，即 expand interactions）简化了把多个多分类变量扩展为成套的虚拟变量和交互变量、并把它们用作回归或者其他模型的解释变量。例如，在数据 *student2.dta*（第 5 章中用过）中，变量 *year* 有 4 种分类，代表大学生入学年数（一年级、二年级，等等）。我们可以自动生成一套由三个虚拟变量组成的变量组，只需键入：

```
. xi, prefix(ind) i.year
```

这三个新的虚拟变量被命名为 *indyear_2*，*indyear_3* 和 *indyear_4*。**prefix( )** 选项用来指定新的虚拟变量命名时所用的前缀。如果我们只是简单键入：

```
. xi i.year
```

而不给出任何 **prefix( )** 选项，那么新的虚拟变量将被命名为 *_Iyear_2*、*_Iyear_3* 和 *_Iyear_4*（并且任何先前计算的变量如果和它们名字相同，将被新的变量覆盖）。要是键入：

```
. drop _I*
```

则是采用通配符 * 来删除所有名字以 *_I* 开头的变量。

xi 在创建虚拟变量时默认地省略分类变量的最小值，但是这可以加以控制。键入命令：

```
. char _dta[omit] prevalent
```

将导致此后的 **xi** 命令自动省略最普遍的类别（注意所用的是方括号）。**char _dta[ ]** 中的偏好是和数据存储在一起的；要想恢复默认设置，键入：

```
. char _dta[omit]
```

键入：

```
. char year[omit] 3
```

将省略 *year* = 3 这一类。要恢复默认设置，键入：

```
. char year[omit]
```

xi 还可以创建涉及两个分类变量的交互项，或者是涉及一个分类变量和一个测量变量的交互项。例如，我们可以创立 *year* 和 *gender* 的交互项，只需键入：

```
. xi i.year*i.gender
```

根据 *year* 的四个类别和 *gender* 的两个类别，**xi** 命令创建 7 个新变量，即 4 个虚拟变量以及 3 个交互项。因为它们的名字都以 *_I* 开头，我们可以使用通配符 *_I* * 并用命令 **describe** 来描述这些变量：

```
. describe _I*

                storage  display     value
variable name     type   format      label      variable label
-------------------------------------------------------------------------------
_Iyear_2          byte   %8.0g                  year==2
_Iyear_3          byte   %8.0g                  year==3
_Iyear_4          byte   %8.0g                  year==4
_Igender_1        byte   %8.0g                  gender==1
_IyeaXgen_2_1     byte   %8.0g                  year==2 & gender==1
_IyeaXgen_3_1     byte   %8.0g                  year==3 & gender==1
_IyeaXgen_4_1     byte   %8.0g                  year==4 & gender==1
```

要创建分类变量 *year* 和测量变量 *drink*（33 级饮酒行为量表）之间的交互项，只需键入：

```
. xi i.year*drink
```

于是出现了 6 个新变量：3 个虚拟变量来表示 *year*，3 个交互项来代表每一个 *year* 虚拟变量乘以 *drink*。例如，对一个大学二年级学生来说，*_Iyear2* = 1，并且 *_IyeaXdrink_2* = 1 × *drink* = *drink*[8]。对一个三年级学生来说，*_Iyear_2* = 0，并且 *_IyeaXdrink_2* = 0 × *drink* = 0；同样地，*_Iyear_3* = 1，有 *_IyeaXdrink_3* = 1 × *drink* = *drink*，如此等等。

---

[8]【译注：这个交互项名称中的 year 被 Stata 将字母 r 自动简化掉了，其中的 X 代表乘号。】

```
. describe _Iyea*

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
_Iyear_2        byte   %8.0g                  year==2
_Iyear_3        byte   %8.0g                  year==3
_Iyear_4        byte   %8.0g                  year==4
_IyeaXdrink_2   float  %9.0g                  (year==2)*drink
_IyeaXdrink_3   float  %9.0g                  (year==3)*drink
_IyeaXdrink_4   float  %9.0g                  (year==4)*drink
```

xi 的真正方便之处在于它在回归或其他模型拟合命令中自动创建虚拟变量和交互项。例如,将变量 *gpa*(学生的大学平均等级分)对 *drink* 和一套 *year* 的虚拟变量进行回归,就简单地键入:

```
. xi: regress gpa drink i.year
```

这一命令按以上描述的规则自动创建必要的虚拟变量。同样,将 *gpa* 对 *drink*、*year* 以及 *drink* 和 *year* 之间的交互项进行回归,键入:

```
. xi: regress gpa drink i.year*drink

i.year            _Iyear_1-4          (naturally coded; _Iyear_1 omitted)
i.year*drink      _IyeaXdrink_#       (coded as above)

      Source |       SS       df       MS              Number of obs =     218
-------------+------------------------------           F(  7,   210) =    3.75
       Model |  5.08865901     7  .726951288           Prob > F      =  0.0007
    Residual |  40.6630801   210  .193633715           R-squared     =  0.1112
-------------+------------------------------           Adj R-squared =  0.0816
       Total |  45.7517391   217  .210837507           Root MSE      =  .44004

------------------------------------------------------------------------------
         gpa |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       drink |  -.0285369   .0140402    -2.03   0.043    -.0562146   -.0008591
    _Iyear_2 |  -.5839268    .314782    -1.86   0.065    -1.204464    .0366107
    _Iyear_3 |  -.2859424   .3044178    -0.94   0.349    -.8860487    .3141639
    _Iyear_4 |  -.2203783   .2939595    -0.75   0.454     -.799868    .3591114
       drink |  (dropped)
_IyeaXdrin~2 |   .0199977   .0164436     1.22   0.225    -.0124179    .0524133
_IyeaXdrin~3 |   .0108977    .016348     0.67   0.506    -.0213297     .043125
_IyeaXdrin~4 |   .0104239    .016369     0.64   0.525    -.0218446    .0426925
       _cons |   3.432132   .2523984    13.60   0.000     2.934572    3.929691
------------------------------------------------------------------------------
```

命令 **xi**:还可以应用于诸如 **logistic**(第 10 章)等其他模型拟合程序之前。通常,它允许我们不用首先创建实际的虚拟变量和交互项就而将如下的解释变量(右侧)包括进来:

i.*catvar*	创建 *j* -1 个虚拟变量来代表 *catvar* 的 *j* 个类别。
i.*carvar1* * **i**.*catvar2*	创建 *j* -1 个虚拟变量来代表 *catvar1* 的 *j* 个类别;根据 *catvar2* 的 *k* 个类别创建 *k* -1 个虚拟变量;并创立 (*j* -1)(*k* -1)个交互项(为虚拟变量 × 虚拟变量)。
i.*catvar* * *measvar*	创建 *j* -1 个虚拟变量来代表 *catvar* 的 *j* 个类别,同时创建 *j* -1 个变量来表示分类变量和测量变量之间的交互作用(虚拟变量 × *measvar*)。

在任何 **xi** 命令执行后,这些新变量将保留在数据中。

逐步回归

随着我们前面在 *states.dta* 中加入虚拟变量项，我们获得了许多 *csat* 的可能解释因素。这导致了过度复杂的模型，其中一些回归系数与零之间并无统计上的显著差别。

```
. regress csat expense percent income college high reg1 reg2
      reg2perc reg3

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  9,    40) =   49.51
       Model |  195420.517     9  21713.3908           Prob > F      =  0.0000
    Residual |   17540.863    40  438.521576           R-squared     =  0.9176
-------------+------------------------------           Adj R-squared =  0.8991
       Total |   212961.38    49  4346.15061           Root MSE      =  20.941

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |  -.0022508   .0041333    -0.54   0.589    -.0106045     .006103
     percent |   -2.93786   .2302596   -12.76   0.000    -3.403232   -2.472488
      income |  -.0004919   .0010255    -0.48   0.634    -.0025645    .0015806
     college |   3.900087   1.719409     2.27   0.029     .4250318    7.375142
        high |   2.175542   1.171767     1.86   0.071     -.192688    4.543771
        reg1 |  -33.78456   9.302983    -3.63   0.001    -52.58659   -14.98253
        reg2 |  -143.5149   101.1244    -1.42   0.164    -347.8949    60.86509
    reg2perc |   2.506616   1.404483     1.78   0.082    -.3319506    5.345183
        reg3 |  -8.799205   12.54658    -0.70   0.487    -34.15679    16.55838
       _cons |   839.2209   76.35942    10.99   0.000     684.8927     993.549
------------------------------------------------------------------------------
```

现在，我们试图来简化模型，首先剔除具有最高 t 概率的自变量（*income*，P = 0.634），接着重新拟合模型以决定再剔除哪些因素。通过这种反向淘汰的过程，我们寻求更加简约的模型，即一个既比较简单而又能拟合较好的模型。理想情况下，在保留或淘汰变量的过程中，应该既注重其统计结果又注重其实际意义或理论意义。

对匆忙的分析人员来说，逐步法提供了模型选择的自动化之路。它们依照预先设立的统计标准，要么是从复杂模型中删除一些自变量，要么是给一个简单的模型加上一些自变量。逐步法并不在其选择过程中考虑变量的实际意义或者理论意义，也不评价和解决每一步模型中可能产生的缺点。不过，尽管有这些缺点，逐步法满足了某些实际需求并得到了广泛使用。

为了完成反向淘汰，我们可以下达 **sw regress** 命令，它将从我们纳入的所有可能的自变量中根据所需的最大 P 值为标准来进行取舍。将保留概率（P-to-retain）设为 **pr(.05)** 可以确保仅有那些系数在 0.05 水平显著不等于 0 的自变量被保留在模型中。

sw regress 首先剔除 *income*，接下来是 *reg3*，然后为 *expense*，最后是 *reg2*，才取得了最终模型。虽然最终模型已经删除了 4 个自变量，但它和原先的模型相比具有几乎同样的 R^2（0.911 8 对比 0.917 6），而其 R_a^2 则反而更高（0.901 8 对比 0.899 1）。

```
. sw regress csat expense percent income college high reg1 reg2
      reg2perc reg3, pr(.05)
```

```
                        begin with full model
p = 0.6341 >= 0.0500  removing income
p = 0.5273 >= 0.0500  removing reg3
p = 0.4215 >= 0.0500  removing expense
p = 0.2107 >= 0.0500  removing reg2

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  5,    44) =   91.01
       Model |  194185.761     5  38837.1521           Prob > F      =  0.0000
    Residual |  18775.6194    44  426.718624           R-squared     =  0.9118
-------------+------------------------------           Adj R-squared =  0.9018
       Total |   212961.38    49  4346.15061           Root MSE      =  20.657

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        reg1 |  -30.59218   8.479395    -3.61   0.001    -47.68128   -13.50309
     percent |  -3.119155   .1804553   -17.28   0.000    -3.482839   -2.755471
    reg2perc |   .5833272   .1545969     3.77   0.000     .2717577    .8948967
     college |   3.995495   1.359331     2.94   0.005     1.255944    6.735046
        high |   2.231294   .8178968     2.73   0.009     .5829313    3.879657
       _cons |    806.672   49.98744    16.14   0.000     705.9289    907.4151
------------------------------------------------------------------------------
```

如果我们采用纳入 *P* 值(*P*-to-enter),例如,**pe(.05)**,代替保留 *P* 值,**sw regress** 将执行(向一个“空”模型即只有常数的模型)做正向纳入自变量,而不是反向剔除。其他的逐步法选项还包括分层选择以及在模型中锁定某些自变量。例如,下面的命令指定了第一项(*x1*)应该被锁定在模型中,因此它并不服从可能的剔除。

```
. sw regress y x1 x2 x3, pr(.05) lockterm1
```

下面的命令要求正向纳入任何在 0.10 水平上显著的解释变量,但是对于变量 *x4*、*x5* 和 *x6* 则需要作为一个整体来处理,或者是一起纳入模型或者是一起排除在外:

```
. sw regress y x1 x2 x3 (x4 x5 x6), pe(.10)
```

下面的命令要求调用分层反向剔除,其中显著性标准定为 $P = 0.20$。

```
. sw regress y x1 x2 x3 (x4 x5 x6) x7, pr(.20) hier
```

选项 **hier** 指定各变量项按序排列,首先考虑移除最后一项(*x7*),如果没有被移除,则停止下来。如果 *x7* 被移除,接下来考虑倒数第二项(*x4 x5 x6*)的移除,如此等等。

除了 **regress**,许多其他的 Stata 命令也有以相似方式工作的逐步法。可以采用的逐步法程序包括如下若干类型:

sw clogit	条件(固定效应)logistic 回归
sw cloglog	最大似然法的互补双对数回归估计
sw cnreg	删截(censored)正态回归
sw glm	一般化线性模型
sw logistic	logistic 回归(输出发生比)
sw logit	logistic 回归(输出系数)
sw nbreg	负二项回归
sw ologit	序次 logistic 回归
sw oprobit	序次 probit 回归
sw poisson	泊松回归
sw probit	probit 回归
sw qreg	分位数回归
sw regress	OLS 回归

sw stcox	Cox 比例风险模型回归
sw streg	参数生存时间模型回归
sw tobit	tobit 回归

键入 **help sw** 可获得关于逐步法选项的细节和原理。

多项式回归

在本章前面,图 6.1 和图 6.2 显示了平均 SAT 综合成绩(*csat*)与高年级高中学生参加测试比例(*percent*)之间存在明显的曲线关系。图 6.6 则显示了在 *percent* 取值较大时 SAT 成绩转向升高,将其视为东北部州特有的情况。那个交互项模型拟合的相当不错($R_a^2 = 0.8327$)。但是后面的图 6.9 所显示的交互项模型的残差对预测值的作图仍然显示出存在一些问题。残差在高预测值和低预测值两头都显示出增加的趋势。

```
. quietly regress csat reg2 percent reg2perc
. rvfplot, yline(0)
```

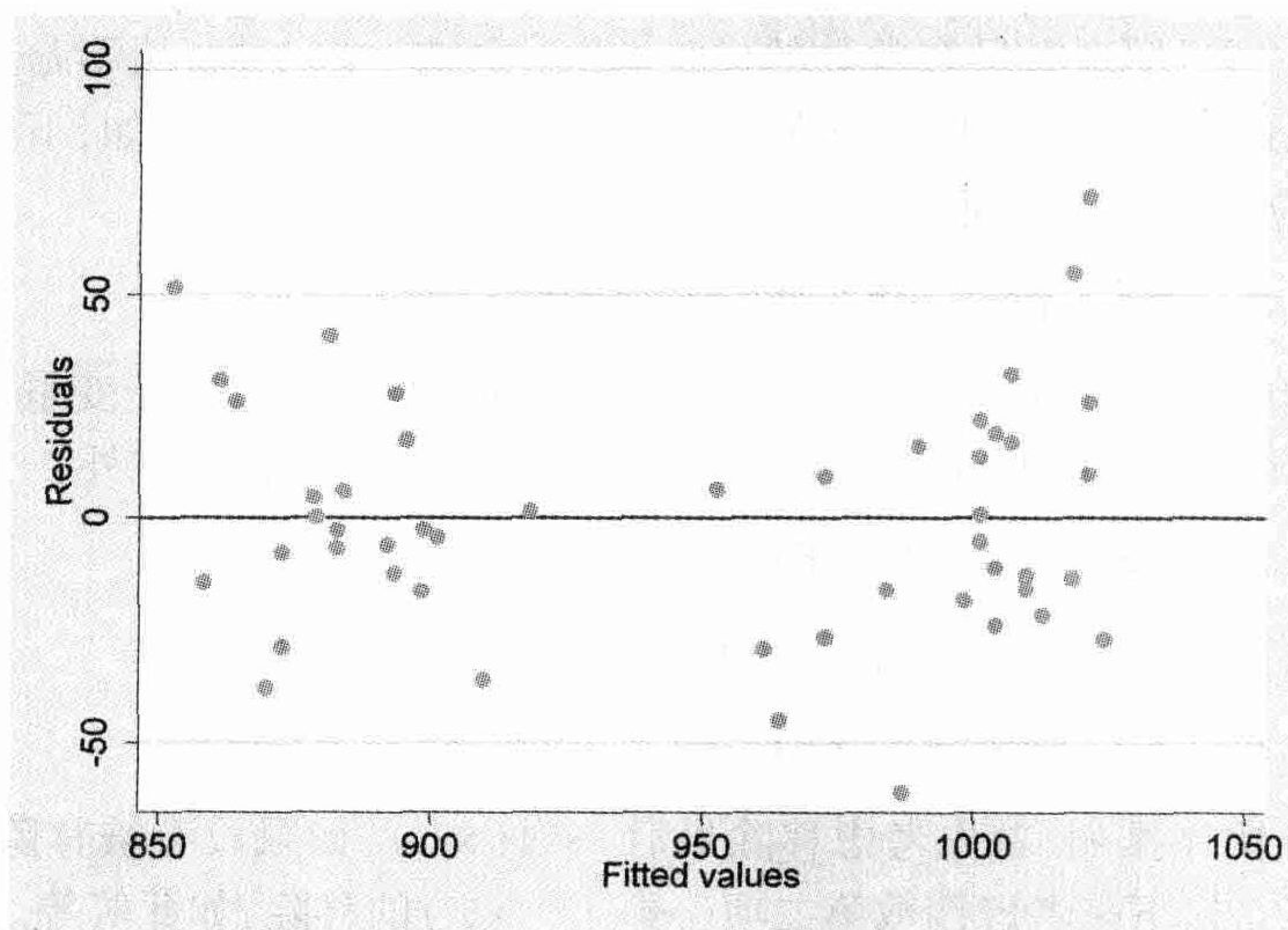


图 6.9

第 8 章介绍了多种处理曲线和非线性回归的技术。这里,"曲线回归"(curvilinear regression)指的是那些包括了对 *y* 变量或者 *x* 变量的非线性转换、但本质上仍是 OLS 的回归(例如,**regress**)。虽然曲线回归根据原始数据进行了曲线模型的拟合,但模型对转换变量还是线性关系。(在第 8 章中也讨论了另一些非线性回归,是应用非最小二乘法来拟合不能通过转换达到线性化的模型)。

有一种被称为多项式回归(polynomial regression)的简单非线性回归,通常能够成功拟合 U 形曲线或者倒 U 形曲线。它的自变量同时包括一个独立的变量以及它的平方项(如果有必要,还可以包括更高次的项)。因为 *csat* 与 *percent* 的关系显示出某种 U 形特征,所以我们创建一个新变量,并使其等于 *percent* 的平方,然后把 *percent* 和 $percent^2$ 都作为 *csat* 的预测变量。图 6.10 描绘了其结果曲线。

```
. generate percent2 = percent^2
. regress csat percent percent2
```

```
      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  2,    48) =  153.48
       Model |  193721.829     2  96860.9146           Prob > F      =  0.0000
    Residual |  30292.6806    48  631.097513           R-squared     =  0.8648
-------------+------------------------------           Adj R-squared =  0.8591
       Total |   224014.51    50   4480.2902           Root MSE      =  25.122

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     percent |  -6.111993   .6715406    -9.10   0.000    -7.462216    -4.76177
    percent2 |   .0495819   .0084179     5.89   0.000     .0326566    .0665072
       _cons |   1065.921   9.285379   114.80   0.000     1047.252    1084.591
------------------------------------------------------------------------------
```

```
. predict yhat2
(option xb assumed; fitted values)

. graph twoway mspline yhat2 percent, bands(50)
      ||  scatter csat percent
      ||  , legend(off) ytitle("Mean composite SAT score")
```

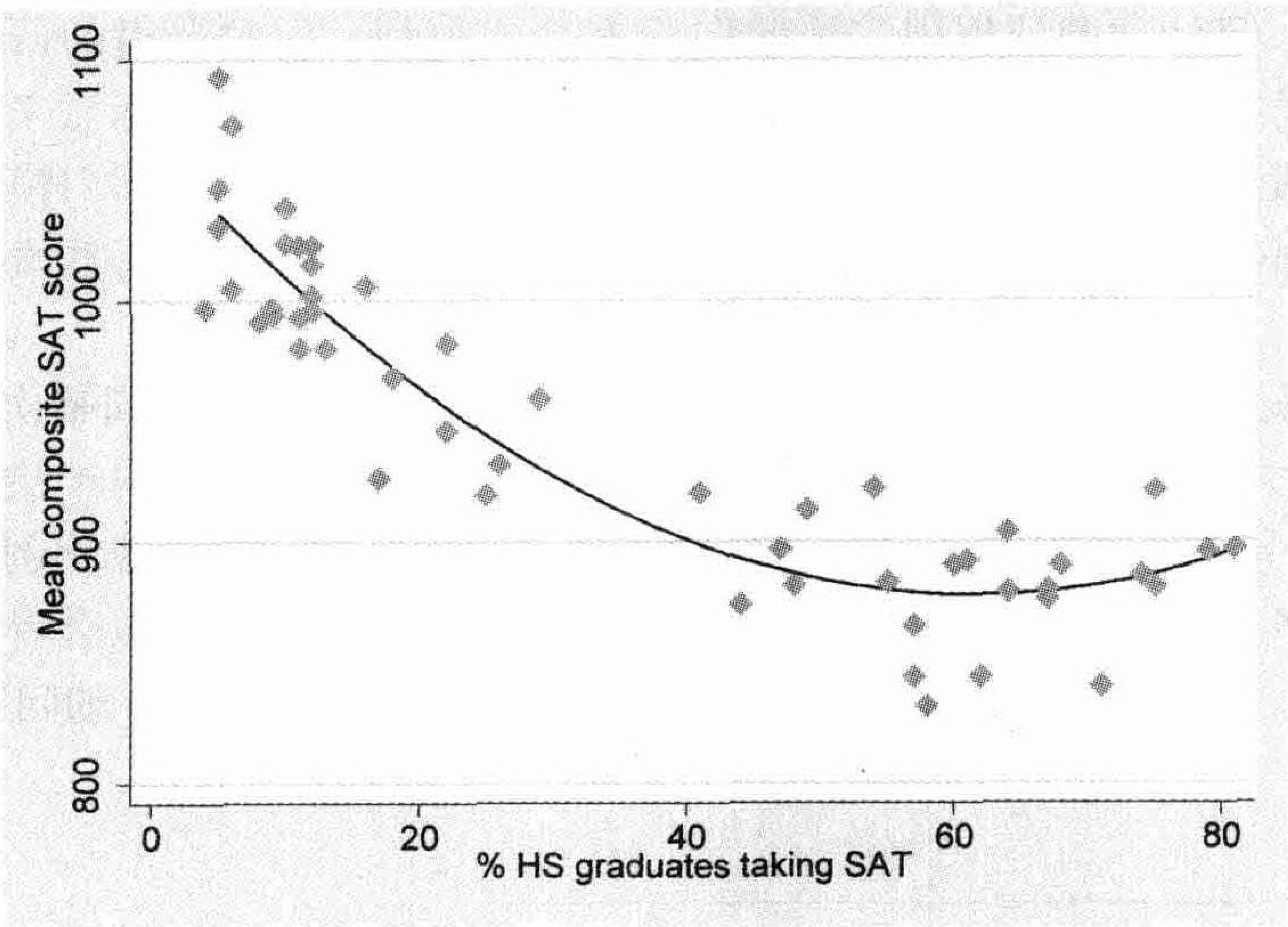


图 6.10

如果我们仅仅想看到这张图,并不需要回归分析,还有一种达到这一目的更快捷的办法。命令 **graph twoway qfit** 可以拟合一个二次多项式回归模型;**qfitci** 还可以画出其置信区间带。例如,一条和图 6.10 相似的曲线可以通过键入以下命令得到:

```
. graph twoway qfit csat percent
      ||  scatter csat percent
```

图 6.10 中的多项式模型拟合数据的情况比起图 6.6 中的交互项模型要稍微好一点(R_a^2 = 0.859 1 相对于 0.832 7)。因为曲线拟合状况在残差对预测值的标绘图(图 6.11)中显得不那么显著,因此采用多项式模型后关于独立同分布误差的通常假定也就显得更容易接受。

```
. quietly regress csat percent percent2

. rvfplot, yline(0)
```

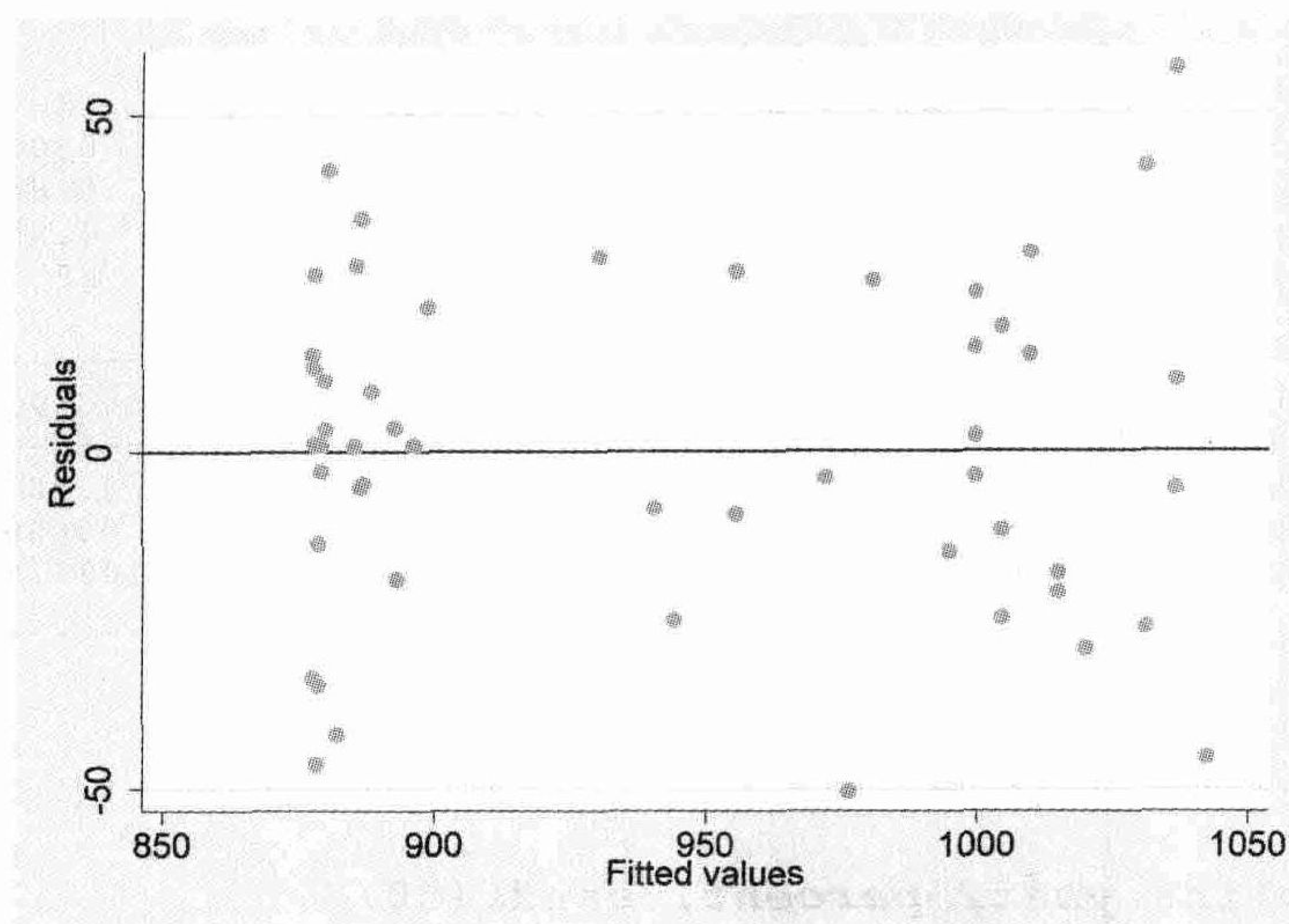


图 6.11

在图 6.7 和图 6.10 中,我们对 SAT 成绩在学生参加率较高水平时的上扬现象就有了两种可选模型。这时,统计学的证据似乎更倾向于多项式模型。但对严肃的研究来说,我们在两个拟合情况相似的模型之间进行选择的时候,应该同时考虑实际意义和统计意义。哪一个模型看起来更有用,或者更有意义?如果只有一个,哪一个回归模型对测试分在学生参加率较高取值时的上扬问题提出了更有实际意义的解释或者和真实情况更为一致?

虽然多项式回归更好地拟合了数据,但它也有重要的统计上的弱点。不同幂次的 x 项互相之间可能高度相关,从而导致多元共线性。此外,多项式回归比较倾向于追随那些 x 取值特大或特小的异常案例,因此一小部分数据点便可以对结果施加不成比例的影响。因为这两个原因,多项式回归的结果有时候会因样本而异,对某个数据拟合得很好但对其他数据则很糟糕。第 7 章将再次讨论这个例子,并用一些工具检查潜在的问题。

面板数据

面板数据(panel data)也叫做横剖时间序列(cross-sectional time series)数据,它由对 *i* 个分析单位或案例在 *t* 个时间点上的观测值组成。《纵向/面板数据参考手册》描述了这种数据的各种分析方法。大多数相关的 Stata 命令都以字母 **xt** 开头;键入 **help xt** 可以得到概览。正如这一文件所说,某些 **xt** 程序要求时间序列数据或者 **tsset** 数据;要得到关于这一步骤的更多信息,参见第 13 章或者键入 **help tsset**。

这一节考虑用命令 **xtreg** 来完成对面板数据做相对简单的线性回归。我们的示例数据 *newfdiv.dta* 包含加拿大纽芬兰省 10 个普查区的数据(Avalon 半岛、Burin 半岛和其他 8 个地方),时间从 1992 年到 1996 年。

图 6.12 将面板数据可视化,画出了其中 9 个普查区中每年报告的犯罪数量变动。第 1 个普查区是 Avalon 半岛,它是目前纽芬兰省最大的一个普查区。用命令 **if cendiv != 1** 将其暂时放到一边会使得图 6.12 中另外的 9 幅图更加易读。这一示例中的 **imargin(l=3 r=3)** 选项要求子图的左右边距宽度等于图形宽度的 3%,从而使子图之间比默认情况分得更开。

```
Contains data from C:\data\newfdiv.dta
  obs:            50                          Newfoundland Census divisions
                                                (source: Statistics Canada)
 vars:             7                          18 Jul 2005 10:28
 size:         2 250 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
cendiv          byte   %9.0g       cd         Census Division
divname         str20  %20s                   Census Division name
year            int    %9.0g                  Year
pop             double %9.0g                  Population, 1000s
unemp           float  %9.0g                  Total unemployment, 1000s
outmig          int    %9.0g                  Out-migration
tcrime          float  %9.0g                  Total crimes reported, 1000s
-------------------------------------------------------------------------------
Sorted by:  cendiv  year
```

```
. list in 1/10
```

	cendiv	divname	year	pop	unemp	outmig	tcrime
1.	Avalon	Avalon Peninsula	1992	259.587	58.56	6556	26.211
2.	Avalon	Avalon Peninsula	1993	261.083	52.23	6449	21.039
3.	Avalon	Avalon Peninsula	1994	259.296	44.81	6907	20.201
4.	Avalon	Avalon Peninsula	1995	257.546	39.35	.	19.536
5.	Avalon	Avalon Peninsula	1996	255.723	38.68	.	21.268
6.	Burin	Burin Peninsula	1992	29.865	9.5	874	1.903
7.	Burin	Burin Peninsula	1993	29.611	9.18	928	1.953
8.	Burin	Burin Peninsula	1994	29.327	8.41	884	1.94
9.	Burin	Burin Peninsula	1995	28.898	7.12	.	2.063
10.	Burin	Burin Peninsula	1996	28.126	6.81	.	1.923

```
. graph twoway connected tcrime year if cendiv != 1,
      by(cendiv, note("")) xtitle("") imargin(left=3 right=3)
```

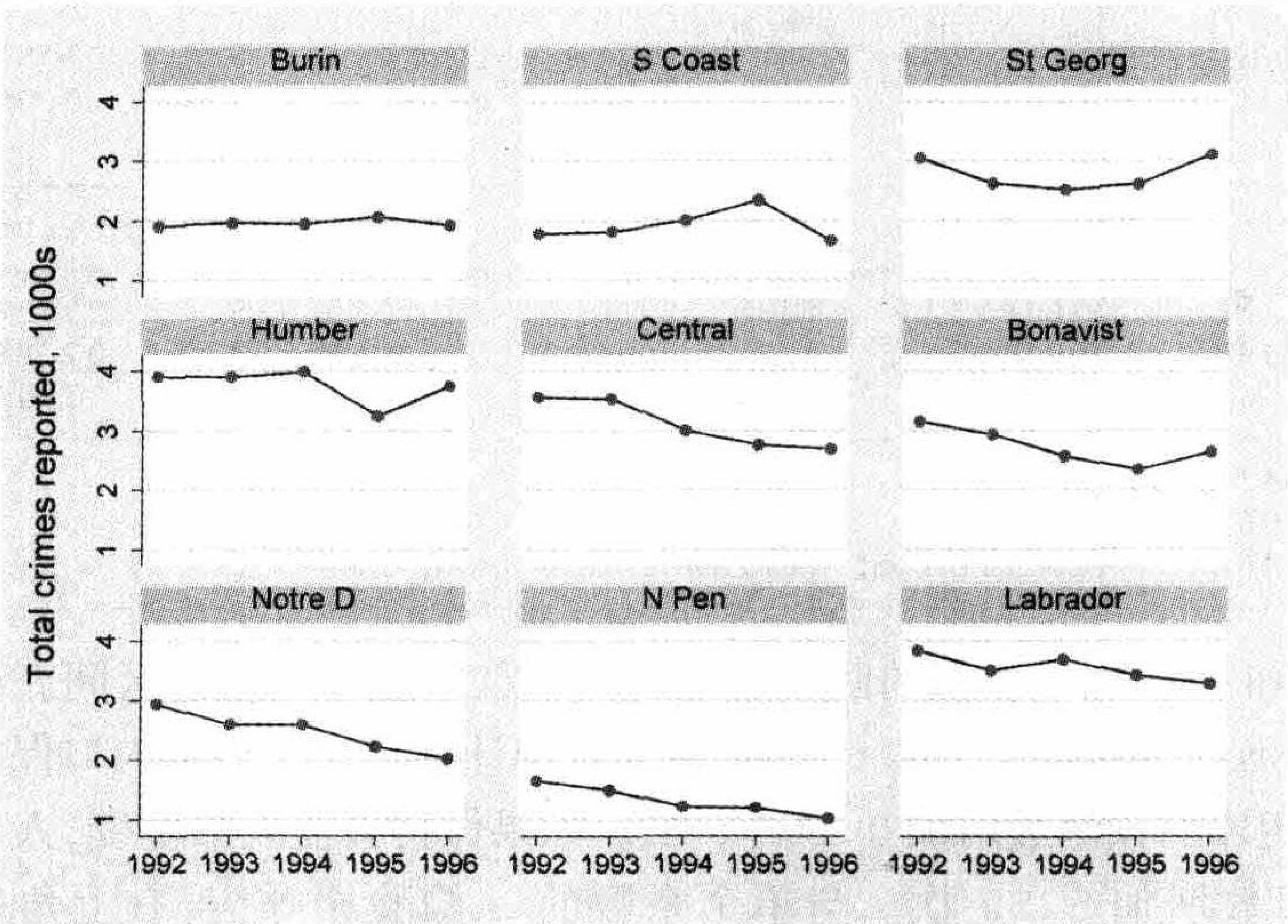


图 6.12

数据中总共包含 50 个观察。但因为这 50 个观察只代表 10 个案例,关于 OLS 的通常假定和其他常规统计方法并不适用。因此,我们需要复杂的误差设置模型,以便允许观察单位上的扰动和单次观察的扰动同时存在。

来考虑 y 对 x 和 w 这两个自变量的回归。OLS 回归估计出回归系数 a、b 以及 c,并

计算出相应的标准误和检验值，这里假设模型具有如下形式：

$$y_i = a + bx_i + cw_i + e_i$$

这里，假定每一观察的残差 e_i 代表了独立同分布下的误差。独立同分布（简标为 i.i.d.）误差假定对于由观察单位的重复测量所组成的面板数据显得不大可能。

一种更为可行的面板数据模型包括两个误差项。一个是对第 *i* 个单位来说是共同的，但是在不同单位之间是不同的（u_i）。第二个则对每一对象 *i* 在时间 *t* 的观察来说都是独特的（e_{it}）。

$$y_{it} = a + bx_{it} + cw_{it} + u_i + e_{it}$$

为了拟合这样一个模型，Stata 需要知道哪一个变量标志第 *i* 个单位，以及哪一个变量是时间标志 *t*。这些可以在一条 **xt** 命令中来实现，或者说是更有效率地将数据作为一个整体。命令 **iis**（意为"*i* 是"）和 **tis**（"*t* 是"）分别指定 *i* 变量和 *t* 变量。对 *newfdiv.dta* 来说，单位是普查区（*cendiv*），时间指标则是 *year*。

```
. iis cendiv
. tis year
. save, replace
```

保存数据将保留对 *i* 和 *t* 的指定。因此，**iis** 和 **tis** 命令在今后的分析中就不需要再重复。设置完这些变量后，我们现在可以拟合一个 *tcrime* 对 *unemp* 和 *pop* 进行回归的随机效应（这意味着共同误差 u_i 被假定为可变的而不是固定的）模型。

```
. xtreg tcrime unemp pop, re

Random-effects GLS regression                   Number of obs      =        50
Group variable (i): cendiv                      Number of groups   =        10

R-sq:  within  = 0.5265                         Obs per group: min =         5
       between = 0.9717                                        avg =       5.0
       overall = 0.9634                                        max =         5

Random effects u_i ~ Gaussian                   Wald chi2(2)       =    705.54
corr(u_i, X)       = 0 (assumed)                Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      tcrime |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       unemp |   .1645266   .0381813     4.31   0.000     .0896925    .2393607
         pop |   .0558997   .0073437     7.61   0.000     .0415062    .0702931
       _cons |  -.7264381    .301522    -2.41   0.016     -1.31741   -.1354659
-------------+----------------------------------------------------------------
     sigma_u |  .34458437
     sigma_e |  .42064667
         rho |  .40157462   (fraction of variance due to u_i)
------------------------------------------------------------------------------
```

xtreg 输出表包含和 OLS 回归相似的回归系数、标准误、*t* 检验和置信区间。在这个示例中，我们看到 *unemp* 的系数（0.164 5）为正值并且统计性显著。如果人口保持不变，每增加一个失业者，犯罪的预测数增加 0.164 5 次。如果保持失业人数不变，人口数每增加 100，预测的犯罪数增加 5.59 次。与单个系数的 *z* 检验相呼应，右上角中的 Wald 卡方检验（$\chi^2 = 705.54, df = 2, P < 0.000\,05$）允许我们拒绝关于 *unemp* 和 *pop* 的系数都等于 0 的联合虚无假设。

输出表还给出关于两个误差项的进一步信息。在表的左下部，我们发现：

sigma_u　　即公共残差 u_i 的标准差

sigma_e　　即独特残差 e_i 的标准差

rho　　由于单位之间差别(即 10 个纽芬兰普查区之间的差异)占未解释方差的比例

$$\mathrm{Var}[u_i]/(\mathrm{Var}[u_i]+\mathrm{Var}[e_{it}])$$

表的左上部给出了三个"R^2"统计量。这些统计量的定义不同于 OLS 中的真实"R^2"。在 **xtreg** 情况下,"R^2"是基于几种 y 的观察值和预测值之间的拟合情况来计算的。

***R*² within**　　单位内解释的变化比例——定义为 y_{it} 值与单位平均数(unit means)的离差($y_{it}-\bar{y}_i$)与预测值和单位平均预测值的离差($\hat{y}_{it}-\hat{y}_i$)之间相关系数的平方。

***R*² between**　　单位间解释的变化比例——定义为单位平均数($\bar{y}_i$)与从自变量预测的单位平均值($\hat{y}_i$)之间相关系数的平方。

***R*² overall**　　总的解释的变化比例——定义为观察值(y_{it})和预测值($\hat{y}_{it}$)之间相关的平方。

我们的示例模型在拟合总的犯罪观察情况时做的很好($R^2=0.96$),并且在普查区平均数的变化方面也做得很好($R^2=0.97$)。但是,对普查区内部变化的预测能力较弱($R^2=0.53$)。

这一示例中的随机效应选项只是以下几种可能选择中的一种。

re　　一般化最小二乘法(GLS)的随机效应(random-effects)估计(默认)

be　　单位间(between)回归估计

fe　　固定效应(fixed-effects)(单位内)回归估计

mle　　最大似然法的随机效应估计

pa　　总体平均(population-averaged)估计

更多的选项和命令语法请查询 **help xtreg** 命令。《纵向/面板数据参考手册》也给出了示例、参考文献和更多的技术细节。

7 回归诊断

数据给了我们不信任所得分析结果的理由吗？我们能够找到更好的方式来设定模型或估计其参数吗？细致的诊断(diagnostics)工作，检查潜在的问题和评价关键假定的合理性，构成了现代数据分析中至关重要的步骤。我们拟合一个初始模型，然后仔细查看结果中可能存在问题的迹象或模型需要改进的方面。前面各章所介绍的诸如散点图、箱线图、正态性检验或者只是排序和列出数据等很多这种一般性方法都有助于发现并修正所存在的问题。Stata 也提供了一套专门为此目的而设计的诊断技术工具箱。

自相关作为常常使时间序列数据回归变复杂的因素并未在本章中涉及。第 13 章的时间序列分析对包括 Durbin-Watson 检验、自相关图、滞后运算和时间序列回归技术等 Stata 的时间序列分析程序库作了介绍。

回归诊断程序可在下面一些菜单选项下找到：

Statistics-Linear regression and related-Regression diagnostics
回归诊断

Statistics-Post-estimation-Predictions, residuals, etc.
取得预测值、残差等

命令示范

本节例举说明的一些命令都假定读者已经先使用 **anova** 或 **regress** 拟合得到了一个模型。命令的结果就是指该模型。这些后续命令(followup commands)有三种基本类型：

1.选项 **predict** 用于创建包含预测值、残差、标准误和影响统计量(influence statistics)等案例统计量信息的新变量。第 6 章曾提到过一些主要的选项。请键入 **help regress** 查看完整的清单。

2.对诸如自相关、异方差性、设定错误或方差膨胀(多元共线性)等统计问题进行诊断检查。请键入 **help regdiag** 查看清单。

3.画出诊断标绘图，诸如附加变量或杠杆作用图、残差对拟合值图、残差对自变量图以及成分对残差图等。同样，请键入 **help regdiag** 取得回归和方差分析诊断图形的完整清单。对分布形状和正态性进行诊断的一般性作图在第 2 章中就已经介绍过；请键入 **help diagplots** 查看那些命令的清单。

命令 **predict** 的选项：

. predict *new*, cooksd

创建一个取值等于 Cook 的距离(Cook's distance)*D* 统计量的新变量，概要描述每个观测案例在多大程度上影响着拟合模型。

. predict *new*, covratio

创建一个取值等于 Belsley、Kuh 和 Welsch 提出的 *COVRATIO* 统计量的新变量。*COVRATIO* 统计量测量第 *i* 个案例对估计系数的方差协方差矩阵的影响。

. predict *DFx1*, dfbeta(*x1*)

创建一个取值等于 *DFBETA* 案例统计量的变量，这里的 *DFBETA* 测量的是每个观测案例对预测变量 *x1* 的系数产生多大影响。用命令 **dfbeta** 可以更方便地完成同样的事情，对于本例，它会自动将作为结果的统计量命名为 *DFx1*：

. dfbeta *x1*

要想对模型中所有预测变量创建完整的一套 *DFBETA*，简单键入 **dfbeta** 即可，无需任何说明。

. predict *new*, dfits

创建 *DFITS* 案例统计量，该统计量也是概要描述每个观测案例对拟合模型的影响(类似于 Cook 的距离 *D* 和 Welsch 的 *W* 统计量)。

诊断检验：

. dwstat

计算一阶自相关的 Durbin-Watson 检验。第 13 章给出这一命令和其他时间序列程序示例。还可见：

help durbina　　Durbin-Watson 的 *h* 统计量

help bgodfrey　　Breusch-Godfrey 的 LM 统计量(拉格朗日乘数，Lagrange multiplier)

. hettest

执行 Cook 和 Weisberg 的异方差性检验。如果我们有理由怀疑异方差分布是特定预测变量 *x1* 的函数，我们可以通过键入 **hettest x1** 把焦点集中在该预测变量上。

. ovtest, rhs

执行对遗漏变量的 Ramsey 回归设定错误检验(*RESET*)。选项 **rhs** 要求使用右侧变量的幂，而不是按预测值 *y* 的幂(默认)。

. vif

计算用于检验多元共线性的方差膨胀因子(variance inflation factors)。

诊断标绘图：

. acprplot *x1*, mspline msopts(bands(7))

在对非线性关系进行检查时，建构一幅扩展分量加残差标绘图(augmented component-plus-residual plot，也称作扩展偏残差标绘图)常常比 **cprplot**(分量加残差图)更佳。选项 **mspline msopts(bands(7))** 要求用线段(line segments)将七个垂直波段的交叉中位数连接起来。作为替代，我们可以通过选项

lowless lsopts(bwidth(.5))要求一条波段宽度为0.5的lowess修匀曲线。

. **avplot** ***x1***

建构一幅附加变量标绘图(added-variable plot,也称作偏回归或杠杆作用图),显示按其他 *x* 变量来调整的 *y* 和 *x1* 之间的关系。此类标绘图有助于发现特异值和影响案例。

. **avplots**

根据最近的 **anova** 或 **regress** 结果,画出所有的附加变量标绘图,并将它们显示在一幅图中。

. **cprplot** ***x1***

建构一幅分量加残差的标绘图(component-plus-residual plot,也称作偏残差标绘图),显示 *y* 和预测变量 *x1* 之间调整的关系。此类标绘图有助于检查数据的非线性关系。

. **lvr2plot**

建构一幅杠杆作用对残差平方的标绘图(leverage-versus-squared-residual plot,也称作L-R标绘图)。

. **rvfplot**

画出残差对 *y* 拟合(预测)值的标绘图(the residuals versus the fitted values of y)。

. **rvpplot** ***x1***

画出预测变量 *x1* 每一取值上的残差。

SAT分数的重新回归

诊断技术一直被当作"回归批评"工具,因为它们有助于我们对回归模型存在的可能不足和可以改进的方面进行检查。本着这一精神,我们现在重新回到第6章中美国各州学术能力测验(SAT)的回归。一个包含三个预测变量的模型解释了州平均SAT分数方差的大约92%。预测变量包括 *percent*(参加测试的高中毕业生的比例)、*percent2*(*percent* 的平方)和 *high*(成人中取得高中文凭的比例)。

. **generate** ***percent2*** **=** ***percent^2***

. **regress** ***csat percent percent2 high***

```
      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  3,    47) =  193.37
       Model |  207225.103     3  69075.0343           Prob > F      =  0.0000
    Residual |  16789.4069    47  357.221424           R-squared     =  0.9251
-------------+------------------------------           Adj R-squared =  0.9203
       Total |   224014.51    50   4480.2902           Root MSE      =   18.90

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     percent |  -6.520312   .5095805   -12.80   0.000    -7.545455   -5.495168
    percent2 |   .0536555   .0063678     8.43   0.000     .0408452    .0664658
        high |   2.986509   .4857502     6.15   0.000     2.009305    3.963712
       _cons |   844.8207   36.63387    23.06   0.000     771.1228    918.5185
------------------------------------------------------------------------------
```

回归方程为:

预测值 $csat = 844.82 - 6.52 percent + 0.05 percent2 + 2.99 high$

图 7.1 中的散点图矩阵描述了这四个变量之间的相互关系。如第 6 章所提到的那样,平方项 *percent2* 为我们的回归模型拟合 *csat* 和 *percent* 之间可见的曲线关系提供了可能。

```
. graph matrix percent percent2 high csat, half msymbol(+)
```

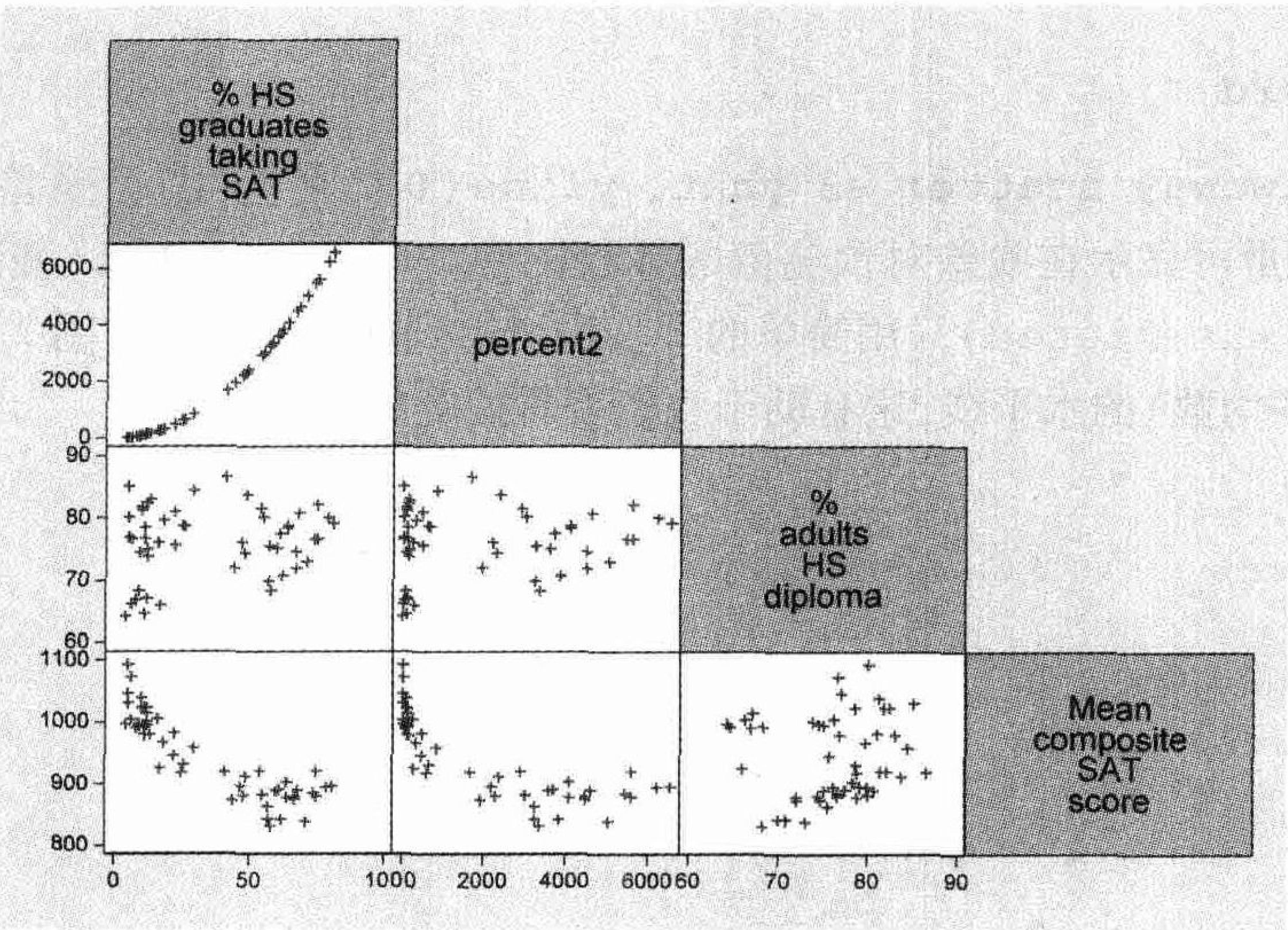


图 7.1

数个回归后续假设检验可对模型设定(model specification)进行检查。遗漏变量检验(omitted variables test)**ovtest** 本质上是 y 对 x 变量以及 y 预测值(先将$\hat{y}$进行标准化使其均值为 0 和方差为 1)的二次幂、三次幂和四次幂进行回归。然后对所有三个$\hat{y}$ 的幂的系数等于零的虚无假设进行 F 检验。如果我们拒绝该虚无假设,那么进一步纳入多项式项来改进模型。就 *csat* 回归而言,我们不必拒绝虚无假设。

```
. ovtest

Ramsey RESET test using powers of the fitted values of csat
       Ho:  model has no omitted variables
                 F(3, 44)  =      1.48
                 Prob > F  =      0.2319
```

异方差性检验(heteroskedasticity test)**hettest** 通过检查标准化残差的平方是否与$\hat{y}$存在线性相关来检查误差方差相等的假定(参见 Cook 和 Weisberg (1994)的讨论与示例)。就 *csat* 回归的结果而言,我们应该拒绝相同方差的虚无假设。

```
. hettest

Cook-Weisberg test for heteroskedasticity using fitted values of csat
     Ho: Constant variance
         chi2(1)      =      4.86
         Prob > chi2  =      0.0274
```

“显著的”异方差性意味着我们的标准误和假设检验可能是无效的。下一节中的图 7.2 显示了为什么会出现这一结果。

诊断标绘图

第 6 章已示范过在运行 **regress** 之后用后续命令 **predict** 创建新变量来保存残差和预测值。为了取得我们就 *csat* 对 *percent*、*percent2* 和 *high* 进行回归所得的这些取值，我们键入两条命令：

```
. predict yhat3
. predict e3, resid
```

通过键入 **graph twoway scatter e3 yhat, yline(0)**，名为 *e3*（残差）和 *yhat3*（预测值）的新变量可显示在残差对预测值的标绘图中。命令 **rvfplot**（残差对预测值，residual-versus-fitted）只用简单的一步也可得到此类标绘图。图 7.2 中包含位于 0 值处（残差平均数）的水平线，这有助于我们查看此类标绘图。

```
. rvfplot, yline(0)
```

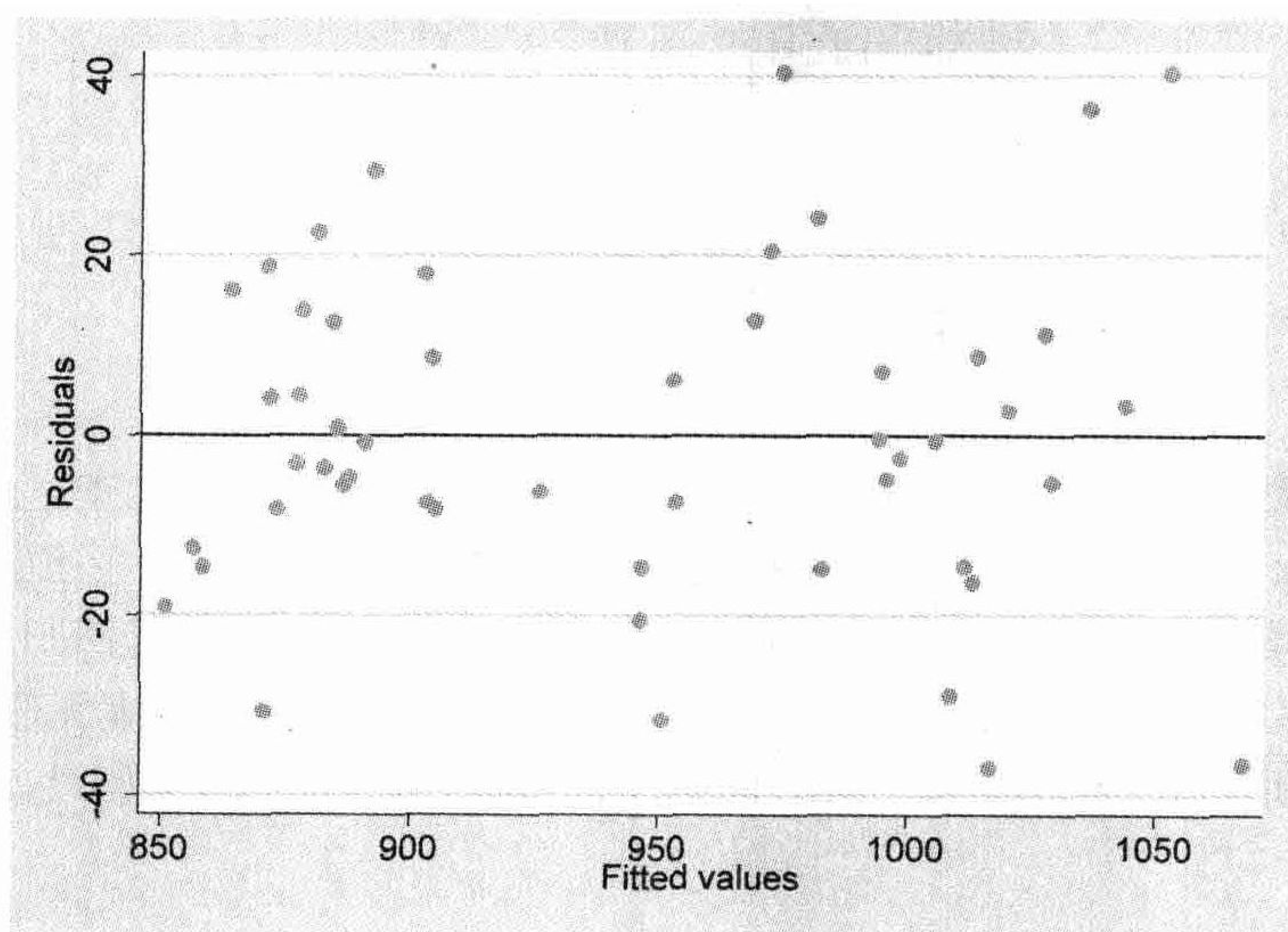


图 7.2

图 7.2 显示了残差围绕在 0 周围对称地分布（对称意味着与正态误差假定相一致），并且没有证据表明存在特异值或曲线关系。但是，残差的离散度对于超过平均水平的 y 预测值的情形似乎略微更大，这就是为什么前面的 **hettest** 会拒绝等方差的假设的原因所在。

残差对拟合值的标绘图只是一种对回归残差的图形概观。为了更详细的研究，我们可以通过一系列"残差对预测变量"的命令分别画出残差对每一预测变量的标绘图。为了画出残差对预测变量 *high* 的标绘图（未显示），请键入：

```
. rvpplot high
```

第 3 章中描述过的单变量图形也可应用于残差分析。比如，我们可以使用箱线图来查看残差体现的特异值或偏态问题，或者使用分位正态标绘图对正态误差假定进行评价。

附加变量标绘图（added-variable plots）是极有价值的诊断工具，它有好几种不同的名称，包括偏回归杠杆作用标绘图（partial-regression leverage plots）、调整的偏残差标绘图（adjusted partial residual plots）或者调整的变

量标绘图(adjusted variable plots)。它们描绘了对其他 x 变量效应进行调整基础上的 y 和某个 x 变量之间的关系。如果我们就 y 对 $x2$ 和 $x3$ 进行回归,并同样就 $x1$ 对 $x2$ 和 $x3$ 进行回归,然后取得每一回归的残差并且在一幅散点图中画出这些残差,我们将得到对 $x2$ 和 $x3$ 进行调整基础上的 y 和 $x1$ 之间关系的附加变量标绘图。命令 **avplot** 会自动执行必要的计算。比如,我们可以画出预测变量 *high* 的调整变量标绘图,仅仅需要键入:

```
. avplot high
```

为了进一步加快处理过程,我们可以键入 **avplots** 以获得一组完整的上述回归中每一预测变量的附加变量标绘图。图 7.3 显示了就 *csat* 对 *percent*、*percent2* 和 *high* 进行回归所得的结果。附加变量标绘图中的直线具有和对应的偏回归系数相等的斜率。比如,图 7.3 左下图直线的斜率等于 2.99,它就是 *high* 的系数。

```
. avplots
```

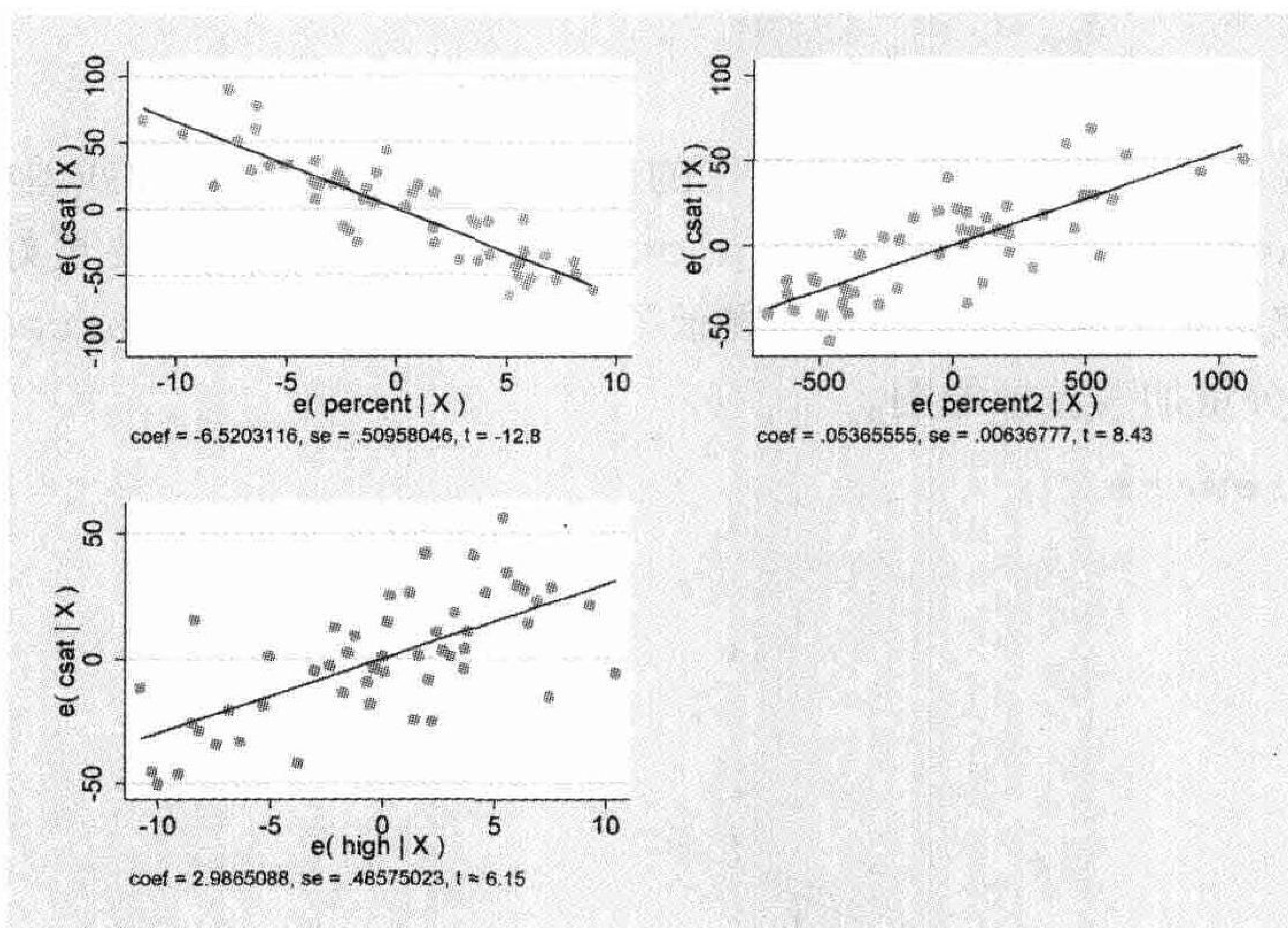


图 7.3

附加变量标绘图有助于发现对回归模型具有不等比例影响的观测案例。在只有一个 x 变量的简单回归中,常规的散点图足以达到这一目的。但是在多元回归中,这种影响的迹象变得更为复杂。在数个 x 变量上具有异常取值组合的一个观测案例可能具有很高的杠杆作用,即更可能对回归造成影响,尽管其单独的 x 的取值本身并不太异常。高杠杆作用的观测案例在附加变量标绘图中显示为在水平方向上远离其他数据的点。但是,我们并没有在图 7.3 中看到此类问题。

如果存在特异值的话,我们可以在附加变量标绘图中的观测案例标志上再添加标签来识别它们是哪些观测案例。正如散点图一样,这可以使用 **mlabel( )** 选项来实现。图 7.4 示范了以州的名称(即字符串变量 *state* 的取值)作为标签。尽管这些标签在数据点分布密集的地方相互重叠,但是个别的特异值仍然能被看出。

由 **cprplot x1** 格式的命令生成的分量加残差标绘图(component-plus-residual plots)采取另一种方式对多元回归进行作图。变量 $x1$ 的分量加残差标绘图画出了在 $x1$ 的取值上每一观测案例的残差及其基于 $x1$ 进行预测的分量,

$$e_i + b_1 x1_i$$

```
. avplot high, mlabel(state)
```

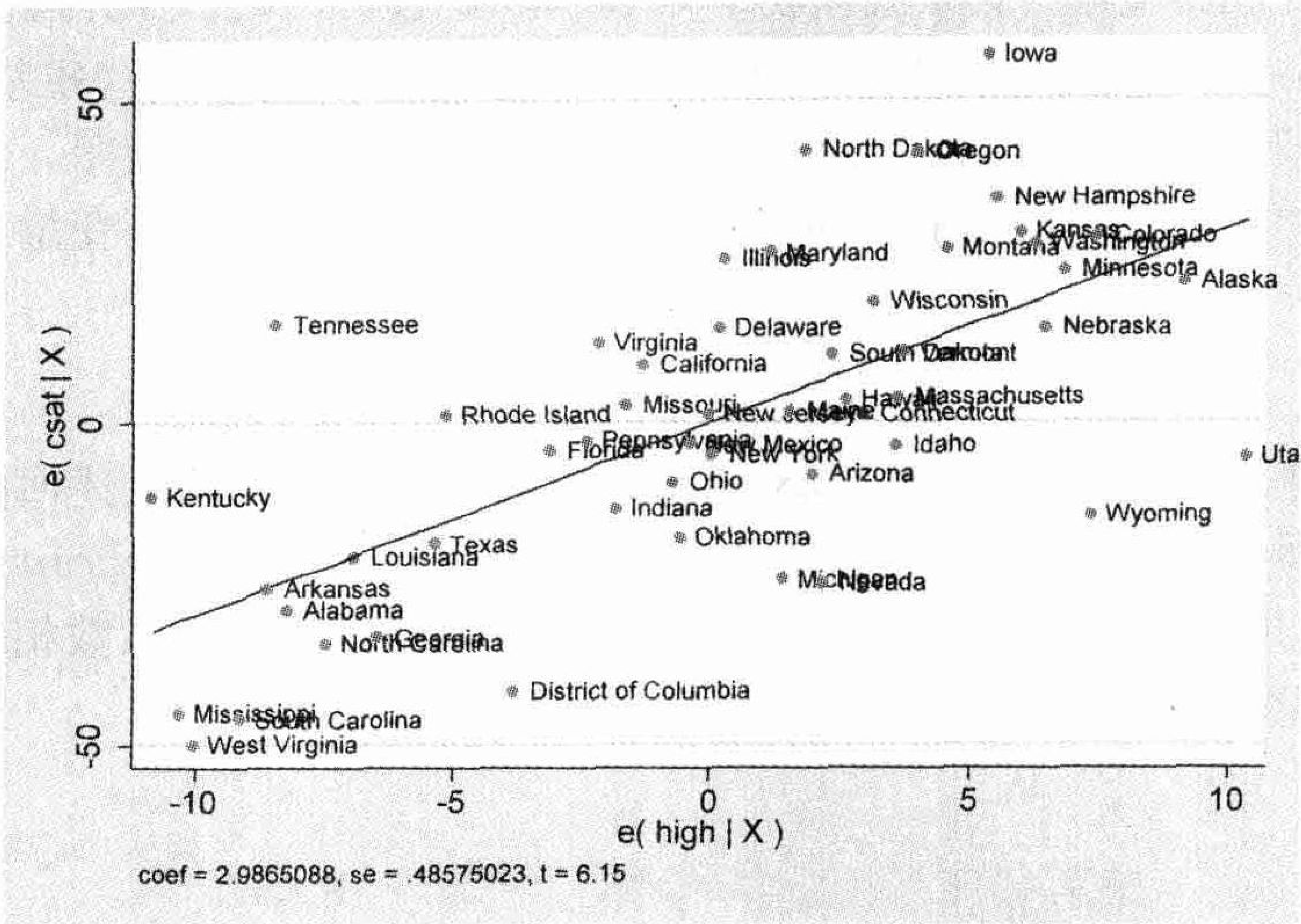


图 7.4

此类标绘图可能有助于检验非线性关系并暗示替代的函数形式。扩展分量加残差标绘图(augmented component-plus-residual plot)(Mallows, 1986)要更好些,尽管两种类型常常似乎都是非决定性的。图 7.5 展示了 *csat* 对 *percent*、*percent2* 和 *high* 进行回归的扩展分量加残差标绘图。

```
. acprplot high, lowess
```

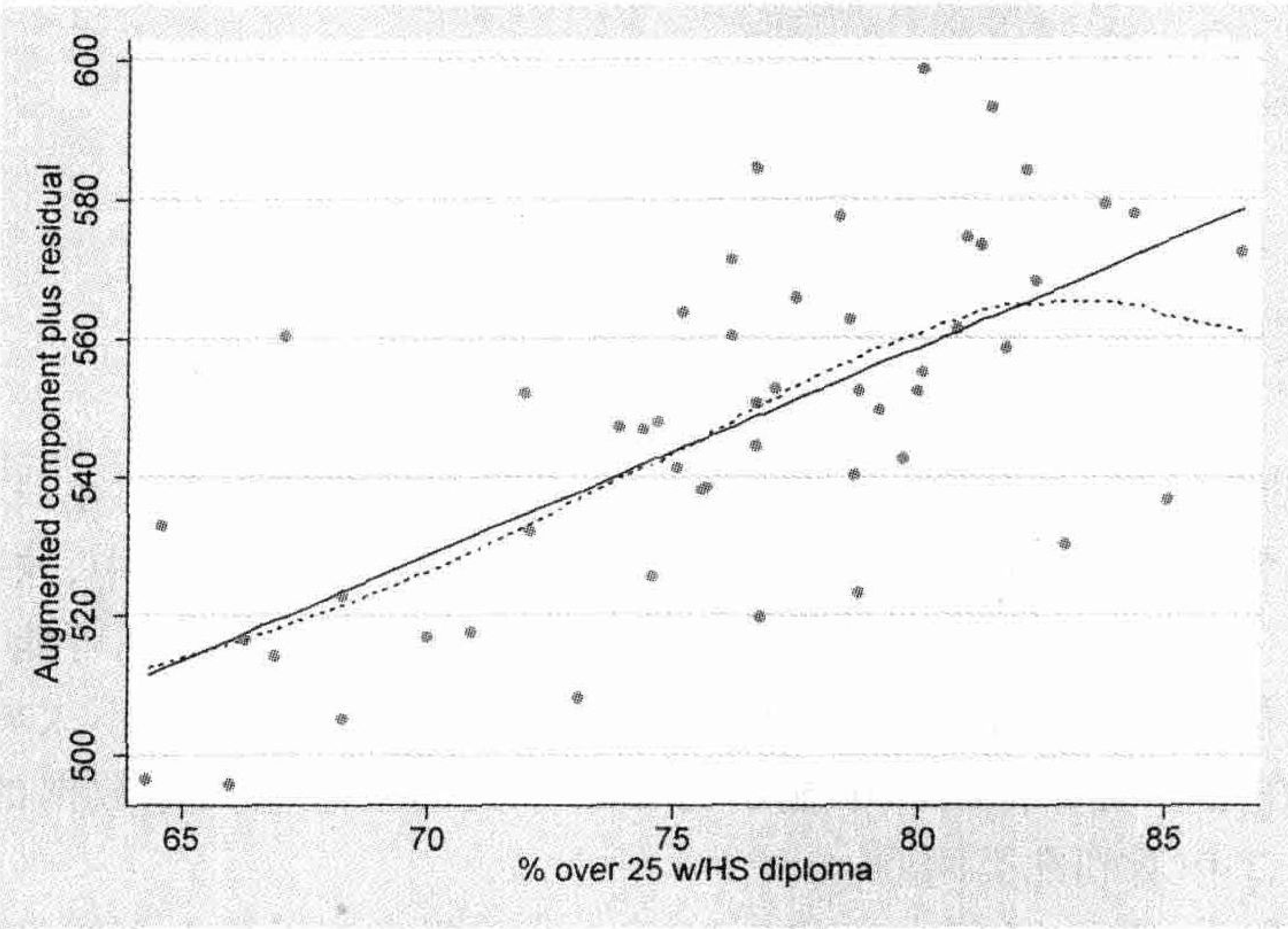


图 7.5

图 7.5 中的直线对应着回归模型。曲线是基于 0.5 的默认波段宽度或一半数据所进行的 lowess 修匀。曲线在右端的下降可作为一种 lowess 假象而忽略它,因为在接近右端时只有少量案例决定其位置(参见第 8 章)。如果 lowess 曲线在靠近中部的地方显示出系统地偏离线性回归模型的弯曲模式,那么我们将有理由怀疑模型的适当性。但是,图 7.5 中分量加残差中位数(component-plus-residuals medians)紧靠回归模型。于是,该图增强了前面我们根据图 7.2 所得到的结论,即当前的回归模型充分解释了原始数据中所有可见的非线性关系(图 7.1),残差中没留下什么明显的非线性关系迹象。

正如其名称所意味的那样，杠杆作用对残差平方的标绘图(leverage-versus-squared-residuals plot)以杠杆作用(即帽子矩阵对角线元素，hat matrix diagonals)对残差的平方作图。图 7.6 展示了就 *csat* 回归的此种图。为了识别个别的特异值，我们以 *state* 的取值在标志上添加标签。选项 **mlabsize(medsmall)** 要求使用“中小”(medium small)字号，这比默认的“小”字号稍大一点。(其他选择的清单，请见 **help testsizestyle**)。尽管州名中的大部分在图 7.6 的左下角混杂在一起，但是少数特异值却得以凸显。

```
. lvr2plot, mlabel(state) mlabsize(medsmall)
```

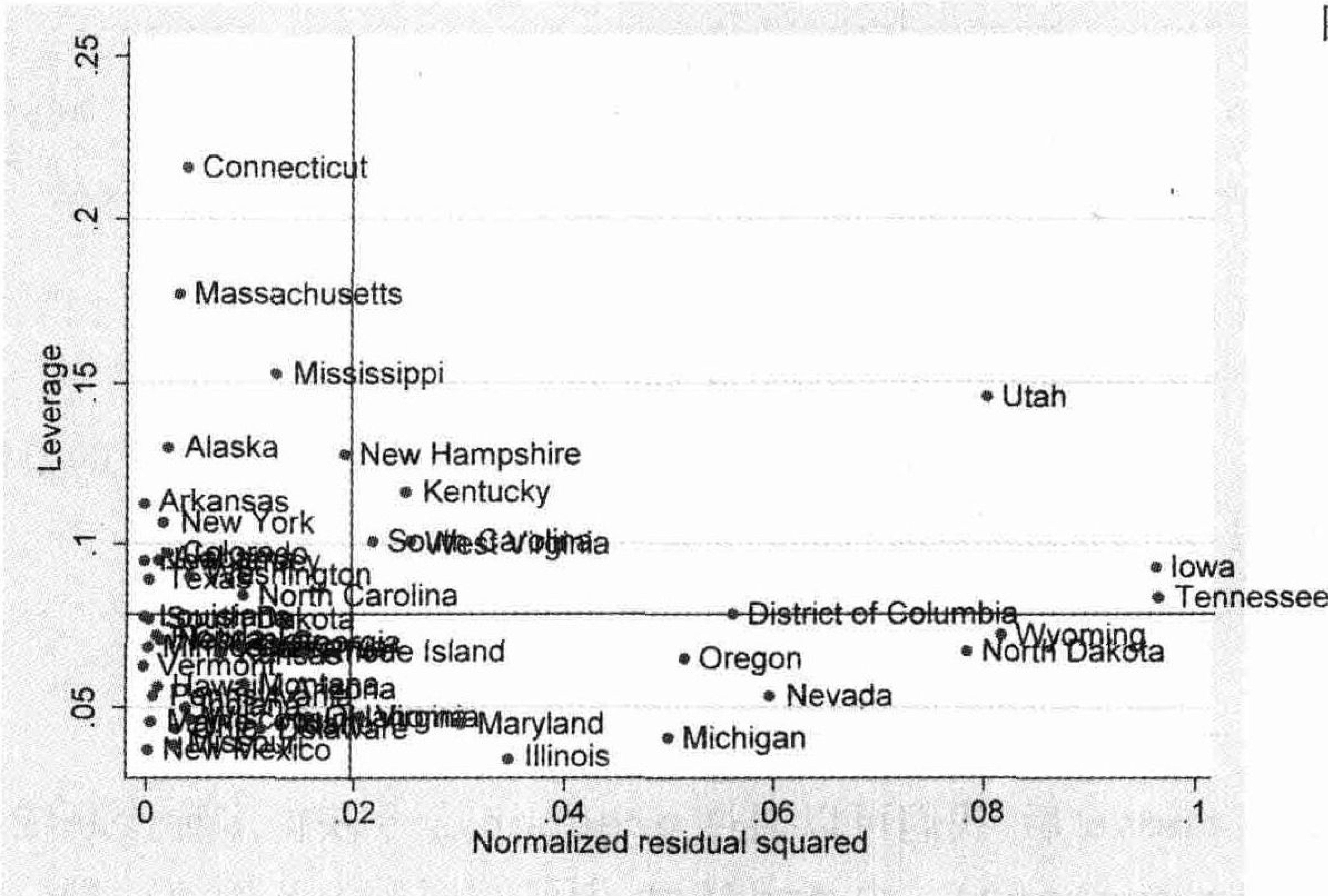


图 7.6

杠杆作用对残差平方标绘图中的直线标示了平均杠杆作用(水平线)和平均的残差平方(垂直线)。杠杆作用告诉我们，基于其 x 取值的特定组合，一个观测案例有多大潜力对回归造成影响。x 的极端取值或异常的组合会导致一个观测案例具有很大的杠杆作用。残差平方较大则表明一个观测案例的 y 值与回归模型预测值极为不同。尽管康涅狄格州、马萨诸塞州和密西西比州(Connecticut, Massachusetts and Mississippi)具有最大的潜在杠杆作用，但是模型对它们的拟合相对较好。(这未必是好事。有时候高杠杆作用案例施加了太大的影响以至于它们支配了回归，因此回归模型肯定对它们拟合得很好。然而，此例并不是这种情况)尽管爱荷华州和田纳西州(Iowa and Tennessee)拟合得不好，但是它们的潜在影响却很小。据分析显示，犹他州(Utah)是一个不但拟合得很差，而且还具有潜在影响的观测案例。我们可以仅列出该州来取得它的值。因为 *state* 是一个字符串变量，所以我们要用英文双引号将“Utah”括起来。

```
. list csat yhat3 percent high e3 if state == "Utah"

     +---------------------------------------------------+
     | csat      yhat3   percent   high           e3 |
     |---------------------------------------------------|
  1. | 1031   1067.712         5   85.1    -36.71239 |
     +---------------------------------------------------+
```

犹他州的学生中只有 5% 参加了学术能力测验，同时该州成年人有 85.1% 为高中毕业。这是一种接近两个 x 变量极端值的异常组合，成为了该州杠杆作用的来源，并导致我们的模型预测出该州学生的平均 SAT 分数比实际平均分数高出 36.7 分。为了准确看

到这一观测案例产生了多大的影响，我们可以使用 Stata 的“不等于”限定条件 != 将 Utah 案例排除后再重新进行回归。

```
. regress csat percent percent2 high if state != "Utah"

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  3,    46) =  202.67
       Model |  201097.423     3  67032.4744           Prob > F      =  0.0000
    Residual |  15214.0968    46  330.741235           R-squared     =  0.9297
-------------+------------------------------           Adj R-squared =  0.9251
       Total |   216311.52    49  4414.52082           Root MSE      =  18.186

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     percent |  -6.778706   .5044217   -13.44   0.000    -7.794054   -5.763357
    percent2 |   .0563562   .0062509     9.02   0.000     .0437738    .0689387
        high |   3.281765   .4865854     6.74   0.000     2.302319     4.26121
       _cons |   827.1159   36.17138    22.87   0.000     754.3067    899.9252
------------------------------------------------------------------------------
```

在样本规模等于 50（而不是 51）的回归中，由于我们删除了一条拟合差（ill-fit）的观测案例，所以所有三个系数略微加强。但是，大体结论并未变化。

Chambers 等（1983）以及 Cook 和 Weisberg（1994）提供了数据分析的诊断标绘图和其他图形方法的更具体的例子和解释。

诊断案例统计量

在运行 **regress** 或 **anova** 后，我们可以通过 **predict** 命令获得多种诊断统计量（参见第 6 章或键入 **help regress**）。由 **predict** 创建的变量为案例统计量（case statistics），意味着它们对于数据中的每一观测案例都有数值。诊断工作通常从计算预测值和残差开始。

其他 **predict** 得到的统计量之间在目的上有一些重叠的地方。许多统计量都试图对每一观测案例在多大程度上影响回归结果进行测量。但是，“影响回归结果”可能指的是不同的事情，比如，对 y 的截距的影响、对特定斜率系数的影响、对所有斜率系数的影响或者是对估计标准误的影响。因此，我们有多种用于测量影响的案例统计量可供选用。

标准化和学生化残差（standardized and studentized residuals）（**rstandard** 和 **rstudent**）有助于识别残差中的特异值，即一些特别背离回归模型的观测案例。学生化残差具有最为简单明了的解释。它们对应着 t 统计量，即要是在回归中纳入一个用 1 表示该条观测案例而用 0 表示所有其他观测案例的虚拟变量时所得到的 t 值。因此，它们检验的是某个特殊观测案例是否显著地改变了 y 的截距。

帽子矩阵对角线元素（hat matrix diagonals）（**hat**）测量了杠杆作用，表明影响回归系数的潜力。当其 x 的取值（或取值的组合）异常时，观测案例会具有高的杠杆作用。

几个其他的统计量测量了对系数的实际影响。*DFBETA* 表明，如果将观测案例 i 从回归中删除，那么 $x1$ 的系数将变化多少个标准误。对于单个预测变量 $x1$，这可以通过两种方式中的任意一种得到：或者通过选项 **predict** 的 **dfbeta(x1)**、或者通过命令 **dfbeta**。

与 *DFBETA* 不同，Cook 的 D 距离（**cooksd**）、Welsch 距离（**welsch**）和 *DFITS* 指

标(**dfits**)都概要描述了观测案例 *i* 对整体回归模型产生了多大影响。换句话说,就是观测案例 *i* 对整套预测值产生了多大影响。*COVRATIO* 测量了第 *i* 个观测案例对估计标准误的影响。下面,我们对所有三个预测变量创建包括 *DFBETA* 在内的整套诊断统计量。请注意,**predict** 会自动地为其创建的变量添加标签,但是 **dfbeta** 不能做到这点。我们从重复最初的回归开始,以确保这些回归后续诊断参照的是恰当的($n=51$)模型。

```
. quietly regress csat percent percent2 high
. predict standard, rstandard
. predict student, rstudent
. predict h, hat
. predict D, cooksd
. predict DFITS, dfits
. predict W, welsch
. predict COVRATIO, covratio
. dfbeta

                       DFpercent:  DFbeta(percent)
                      DFpercent2:  DFbeta(percent2)
                          DFhigh:  DFbeta(high)

. describe standard - DFhigh

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
standard        float  %9.0g                  Standardized residuals
student         float  %9.0g                  Studentized residuals
h               float  %9.0g                  Leverage
D               float  %9.0g                  Cook's D
DFITS           float  %9.0g                  Dfits
W               float  %9.0g                  Welsch distance
COVRATIO        float  %9.0g                  Covratio
DFpercent       float  %9.0g
DFpercent2      float  %9.0g
DFhigh          float  %9.0g

. summarize standard - DFhigh

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    standard |        51   -.0031359    1.010579  -2.099976   2.233379
     student |        51     -.00162    1.032723  -2.182423   2.336977
           h |        51    .0784314    .0373011   .0336437   .2151227
           D |        51    .0219941    .0364003   .0000135   .1860992
       DFITS |        51   -.0107348    .3064762   -.896658   .7444486
-------------+--------------------------------------------------------
           W |        51    -.089723    2.278704  -6.854601    5.52468
    COVRATIO |        51    1.092452    .1316834   .7607449   1.360136
   DFpercent |        51     .000938    .1498813  -.5067295   .5269799
  DFpercent2 |        51   -.0010659    .1370372   -.440771   .4253958
      DFhigh |        51   -.0012204    .1747835  -.6316988   .3414851
```

summarize 为我们显示了每个统计量的最小值和最大值,因此我们可以迅速检查是否有哪一个统计量大到足以引起我们的注意。比如,一些特殊表格可用于确定学生化残差(*student*)绝对值最大的观测案例是否构成一个显著的特异值。作为替代办法,我们也可以使用 Bonferroni 不等式和 t 分布表:如果绝对值 $|t|$在 α/n 水平显著,那么 |*student* |的最大值就会在 α 水平显著。本例中, |*student*|的最大值等于 2.337(Iowa,即爱荷华州),并且 n 为 51。由于爱荷华州在 $\alpha=0.05$ 水平成为一个显著的特

异值(即导致截距显著变化),因此 $t=2.337$ 必定在0.05/51水平显著:

```
. display .05/51
.00098039
```

在给定自由度 $df=n-K-1=51-3-1=47$ 的情况下,Stata 的 **ttail( )**函数可近似计算出 $|t|>2.337$ 的概率为:

```
. display 2*ttail(47, 2.337)
.02375138
```

所得 P 值($P=0.0238$)没有低于 $\alpha/n=0.00098$,因此爱荷华州并不是在 $\alpha=0.05$ 水平显著的特异值。

学生化残差测量了第 i 个观测案例对 y 的截距的影响。Cook 的 D、$DFITS$ 和 Welsch 距离都测量了第 i 个观测案例对模型中所有系数(或者换句话说,对所有 n 个 y 预测值)的影响。为了列出由 Cook 的 D 测量出的5个具有最大影响的观测案例,键入:

```
. sort D
. list state yhat3 D DFITS W in -5/l

              state      yhat3          D        DFITS           W
47.    North Dakota   1036.696   .0705921     .5493086    4.020527
48.         Wyoming   1017.005   .0789454    -.5820746   -4.270465
49.       Tennessee   974.6981    .111718     .6992343    5.162398
50.            Iowa    1052.78   .1265392     .7444486     5.52468
51.            Utah   1067.712   .1860992     -.896658   -6.854601
```

选择条件 **in -5 /l**(注意是小写字母"l")告诉 Stata 只列出从最后一个观测案例开始的倒数5个观测案例。图7.7 展示了一种以图形方式显示影响的方式:通过使用"分析权数"(analytical weight)选项 [**aweight = D**],残差对预测值标绘图中的记号大小将与 Cook 的 D 取值成比例。具有较大正负残差和较高 $csat$ 预测值的五条观测案例凸显出来。

```
. graph twoway scatter e3 yhat3 [aweight = D], msymbol(oh) yline(0)
```

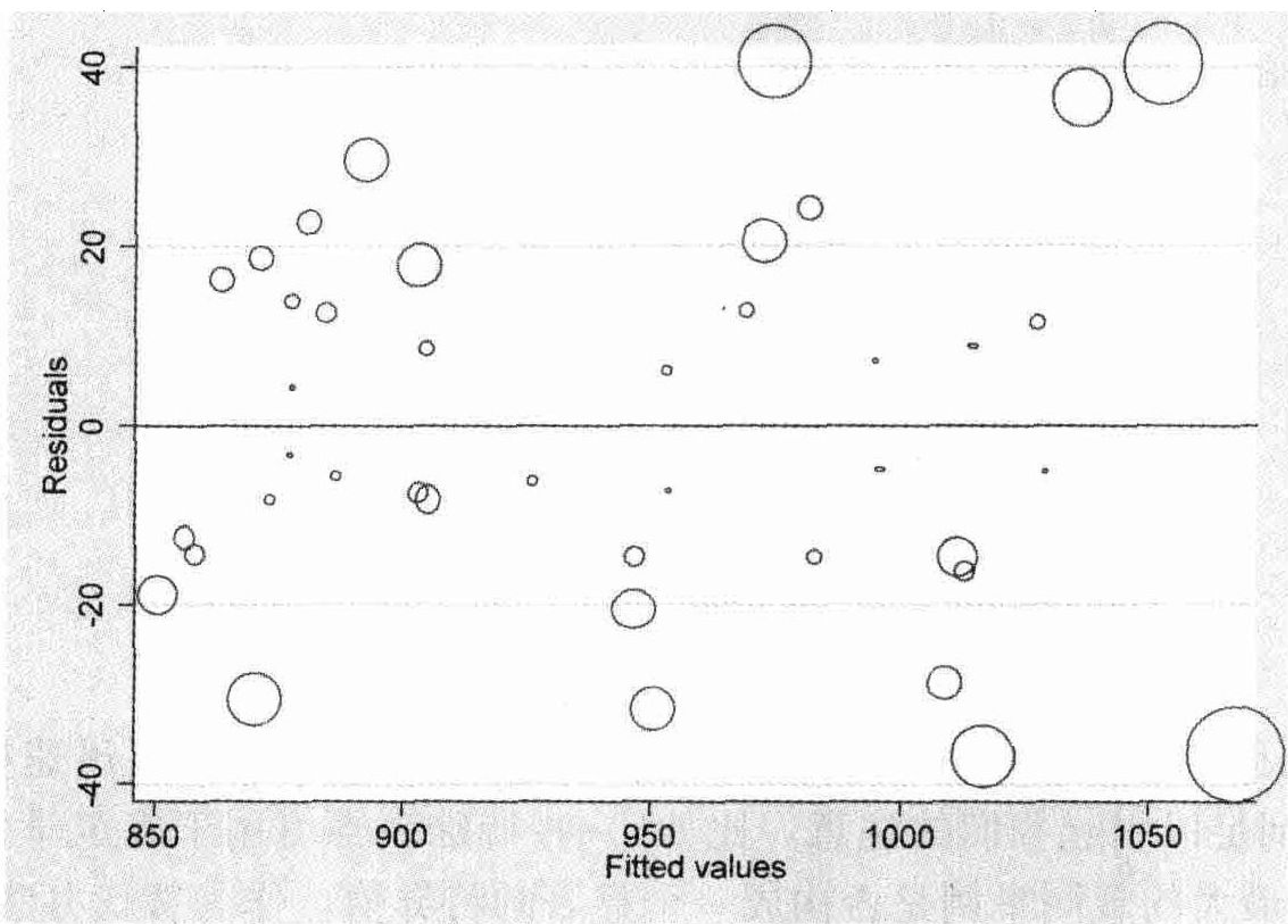


图7.7

尽管具有不同的统计原理,但是 Cook 的 D、Welsch 距离和 $DFITS$ 具有密切联系。实践中它们往往将同样的观测案例标记为影响案例。图7.8 以当前的例子展示了它们

的相似之处。

```
. graph matrix D W DFITS, half
```

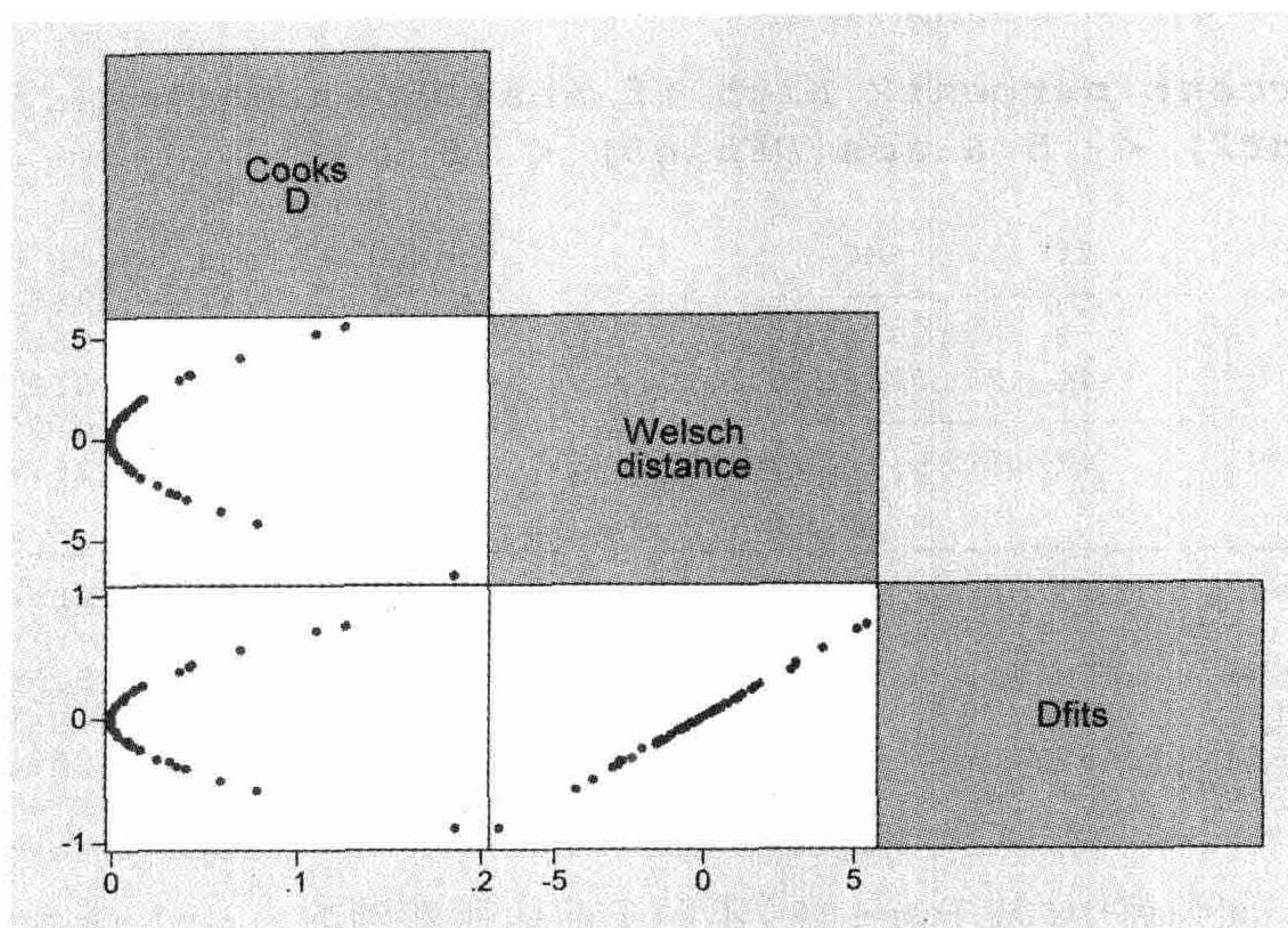


图 7.8

DFBETA 表明每一观测案例对每一回归系数有多大的影响。执行回归之后键入 **dfbeta** 会自动对每一预测变量创建 *DFBETA*。在本例中，它们被命名为 *DFpercent*（即对预测变量 *percent* 的 *DFBETA*）、*DFpercent2* 和 *DFhigh*。图 7.9 采用箱线图画出了它们的分布。

```
. graph box DFpercent DFpercent2 DFhigh, legend(cols(3))
```

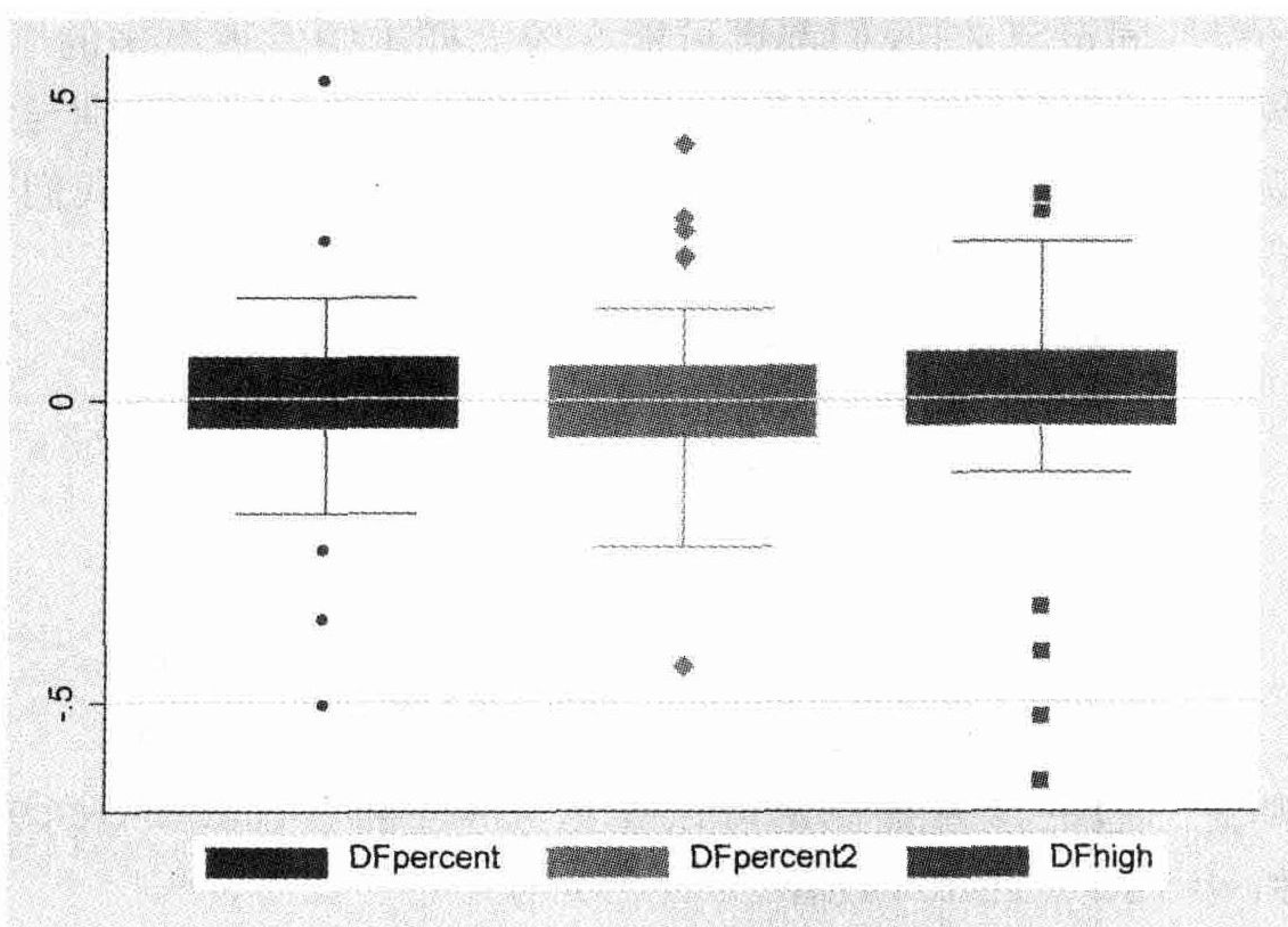


图 7.9

从左到右，图 7.9 依次显示了针对 *percent*、*percent2* 和 *high* 的 *DFBETA* 的分布。（如果着色，我们就更容易区分它们。）每一幅图中的极端值都属于爱荷华州和犹他州（Iowa 和 Utah），它们两者还具有最大的 Cook 的 *D* 值。比如，犹他州的 *DFhigh* = -0.63。这告诉我们，犹他州会导致 *high* 的回归系数比排除该案例时低 0.63 个标准误。类似地，*DFpercent* = 0.53 表明，在犹他州存在的情况下，*percent* 的回归系数比排除该案例时高 0.53 个标准误（因为 *percent* 的回归系数是负的，"更高"意味着更接近于 0）。因此，犹他州弱化了 *high* 和 *percent* 两者表面上的影响。

了解特殊观测案例在多大程度上影响着一个回归的最直接的方式就是在排除那些观测案例的情况下重新进行回归。比如,我们可以排除所有造成各个系数半个标准误(即 *DFBETA* 的绝对值大于等于 0.5)变化的那些州:

```
. regress csat percent percent2 high if abs(DFpercent) < .5 &
      abs(DFpercent2) < .5 & abs(DFhigh) < .5

      Source |       SS       df       MS              Number of obs =      48
-------------+------------------------------           F(  3,    44) =  215.47
       Model |  175366.782     3  58455.5939           Prob > F      =  0.0000
    Residual |  11937.1351    44  271.298525           R-squared     =  0.9363
-------------+------------------------------           Adj R-squared =  0.9319
       Total |  187303.917    47  3985.18972           Root MSE      =  16.471

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     percent |  -6.510868   .4700719   -13.85   0.000    -7.458235     -5.5635
    percent2 |   .0538131    .005779     9.31   0.000     .0421664    .0654599
        high |    3.35664   .4577103     7.33   0.000     2.434186    4.279095
       _cons |   815.0279   33.93199    24.02   0.000     746.6424    883.4133
------------------------------------------------------------------------------
```

仔细查看会发现这一回归表中(基于 $n=48$)不同于前面所看到的 $n=51$ 或 $n=50$ 表中对应部分。但是,我们的主要结论仍然没有改变,即平均的州 SAT 分数可很好地由高中毕业的成年人比例和参加测验的学生比例来非线性地预测。

尽管诊断统计量将注意力放在具有很大影响的观测案例(influential observations)上面,但是它们并没有回答我们是否应当排除那些观测案例的问题。那要求在仔细评估数据和研究背景的基础来作实际决定。本例中,我们没有任何实际理由去排除任何一个州,即使是其中具有最大影响的州也没有根本性地改变我们的结论。

采用任何关于"特异值"的固定定义,我们都更可能在较大样本中发现更多的"特异值"。由于这个原因,依据样本规模调整的分界点(cutoff)有时被推荐用来识别异常案例。基于 n 个观测案例拟合一个具有 K 个系数(包含常数项)的回归模型之后,我们可以更密切地关注那些下列任何一个表达式为真的观测案例:

杠杆作用	leverage $h > 2K/n$
Cook 的 D 统计量	Cook's $D > 4/n$
DFITS 统计量	$DFITS > 2\sqrt{K/n}$
Welsch 的 W 距离	Welsch's $W > 3\sqrt{K}$
DFBETA 统计量	$DFBETA > 2/\sqrt{n}$
COVRATIO 统计量	$\lvert COVRATIO - 1 \rvert \geqslant 3K/n$

这些分界点背后的原理和更一般的诊断统计量可以参见 Cook 和 Weisberg(1982,1994),Belsley、Kuh 和 Welsch(1980)或者 Fox(1991)。

多元共线性

如果预测变量之间存在完全多元共线性(multicollinearity,即存在线性关系),回归方程就会无解。这时,Stata 会向用户提出警告然后以自动删除某个预测变量的方式对此进行处理。较高但不是完全的多元共线性的情况会导致更难以捉摸的问题。当我们加入一个与模型中已有 x 变量存在高度相关的新的 x 变量时,可能存在问题的征兆如下:

1.标准误变得很大,而对应的 t 统计量却更小。

2.系数的数值和符号出乎意料的变化。

3.R^2 虽然很大但回归系数却并不显著。

多元回归试图估计每个 x 变量的独立影响。但是,如果 x 变量中的一个或更多个变量并不具有多少独立变异,那么能支持这样做的信息就很少。以上列出的征兆提示我们,系数估计会变得很不可靠,同时样本或模型中的细微变化就可能彻底改变这些系数估计值。要进一步解决问题就需要确定多元共线性是否真的成为了问题,如果是,又应当如何进行处理。

检查变量之间的相关矩阵未必能够察觉或消除多元共线性。更好的办法是就每一 x 对所有其他 x 变量进行回归。然后我们根据这一回归计算出 $1-R^2$ 来查看第一个 x 变量的方差中与其他 x 变量独立的比例是多少。比如,*high* 的方差中有大约 97% 独立于 *percent* 和 *percent2*:

```
. quietly regress high percent percent2
. display 1 - e(r2)
.96942331
```

回归之后,e(r2)保存着 R^2 的数值。类似的命令还显示 *percent* 方差只有 4% 独立于其他两个预测变量:

```
. quietly regress percent high percent2
. display 1 - e(r2)
.04010307
```

这一有关 *percent* 和 *percent2* 的结果并不令人惊讶。在多项式回归(polynomial regression)或包含交互项的回归中,一些 x 变量可能是直接根据其他 x 变量计算得到。尽管它们的关系严格地讲是非线性的,但是常常又是很接近于线性的,从而造成多元共线性问题。

对于方差膨胀因子(variance inflation factor),回归后续命令 **vif** 可以类似地进行自动计算。这能迅速对多元共线性作简捷的检查:

```
. quietly regress csat percent percent2 high
. vif

    Variable |       VIF       1/VIF
-------------+----------------------
     percent |     24.94    0.040103
    percent2 |     24.78    0.040354
        high |      1.03    0.969423
-------------+----------------------
    Mean VIF |     16.92
```

位于 **vif** 输出表右边的 1/VIF 一列给出了 $1-R^2$ 的数值,它根据每一 x 对其他 x 变量进行回归计算得到,这一点可将 *high*(0.969 423)或 *percent*(0.040 103)的值与我们前面 **display** 的计算值一比较就能看出。也就是说,1/VIF(或 $1-R^2$)告诉我们某个 x 变量的方差独立于所有其他 x 变量的比例是多少。要是比例小,比如,*percent* 和 *percent2* 的 0.04(4% 的独立变异),就表明可能存在问题。一些分析人员对 1/VIF 的值设定了一个被称作容忍度(*tolerance*)的最低水平,并自动排除那些低于他们的容忍度标准的预测变量。

位于 **vif** 表中间的 VIF 列反映了由于该预测变量的纳入所带来的其他变量系数的

方差(及标准误)增加的程度。我们看到,*high* 实际上对其他变量没有影响,但是 *percent* 和 *percent2* 对对方方差的影响很大。VIF 的值提供了系数方差增加的指示,但它并不是对系数方差增加的直接测量。下述命令通过显示 *percent2* 在纳入和不纳入模型时 *percent* 系数的标准误估计值来直接展示了这一影响。

```
. quietly regress csat percent percent2 high
. display _se[percent]
.50958046
. quietly regress csat percent high
. display _se[percent]
.16162193
```

当 *percent2* 纳入模型时,*percent* 的标准误是未纳入 *percent2* 时的3倍:

$$0.509\ 580\ 46\ /0.161\ 621\ 93\ =\ 3.152\ 916\ 6$$

这相当于系数方差上的10倍增加。

方差膨胀多少才算太大? Chatterjee、Hadi 和 Price(2000)建议用以下条件作为多元共线性存在的判断标准:

1.最大的 VIF 大于10;

2.平均的 VIF 大于1。

由于我们最大的 VIF 接近25,同时平均的 VIF 几乎达到了17,因此 *csat* 回归明显地满足这两条标准。问题很棘手,还能做些什么呢,这就是下一个需要考虑的问题。

因为 *percent* 和 *percent2* 紧密相关,因此我们不能像单独估计任一预测变量的作用时那样精确地估计出它们的独立效应。这就是为什么当我们将 *csat* 对 *percent* 和 *high* 的回归与 *csat* 对 *percent*、*percent2* 和 *high* 的多项式回归比较时发现 *percent* 系数的标准误增加了三倍的原因。尽管这有失精确,但是我们仍然可以将所有的系数与零区分开来。而且,多项式回归获得了一个更好的预测模型。考虑到这些原因,这一回归中的多元共线性未必引起了大问题,或者要求某种解决办法。我们可以简单地接受它,就如同其他可接受模型也会有自己的特征一样。

```
. summarize percent

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     percent |        51    35.76471    26.19281          4         81

. generate Cpercent = percent - r(mean)
. generate Cpercent2 = Cpercent ^2
. correlate Cpercent Cpercent2 percent percent2 high csat
(obs=51)

             | Cpercent Cperce~2  percent percent2     high     csat
-------------+------------------------------------------------------
    Cpercent |   1.0000
   Cpercent2 |   0.3791   1.0000
     percent |   1.0000   0.3791   1.0000
    percent2 |   0.9794   0.5582   0.9794   1.0000
        high |   0.1413  -0.0417   0.1413   0.1176   1.0000
        csat |  -0.8758  -0.0428  -0.8758  -0.7946   0.0858   1.0000
```

如果需要予以解决时,一个被称作"对中"(centering)的简单窍门常常可以成功地减少多项式或交互效应模型中的多元共线性。对中就是在创建多项式或乘积项之前

将 x 变量值减去其平均数。减去平均数导致创建的新变量以零为中心分布，并且该新变量与其值的平方项的相关会大大削弱。随后的回归的拟合优度与未对中的回归是一样的。通过减少多元共线性，对中常常(但并不总是)得到更精确的系数估计值，即具有更小的标准误。下面的命令用于创建名 *percent* 的对中变量 *Cpercent*，然后创建名为 *Cpercent2* 的 *Cpercent* 平方项。

尽管 *percent* 和 *percent2* 相互之间几乎完全相关($r = 0.979\ 4$)，但是对中的 *Cpercent* 和 *Cpercent2* 却仅为中度相关($r = 0.379\ 1$)。另外，*percent* 和 *Cpercent* 的相关系数等于1，因为对中只是一种线性转换。但是，涉及 *Cpercent2* 的相关系数与那些涉及 *percent2* 的相关系数很不一样。图 7.10 展示的散点图有助于直接观察这些相关和转换效果。

```
. graph matrix Cpercent Cpercent2 percent percent2 high csat,
      half msymbol(+)
```

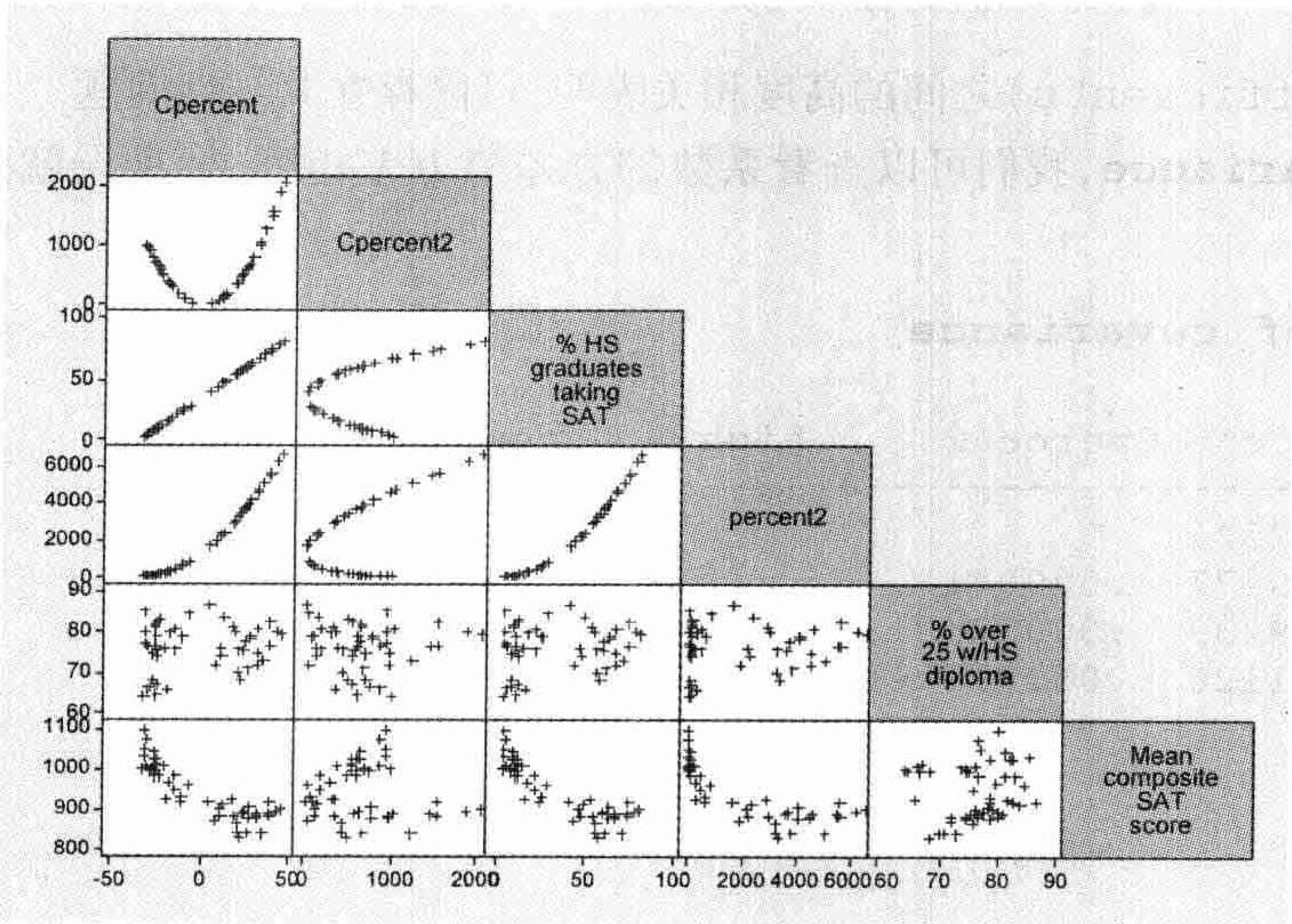


图 7.10

对中之后，模型的 R^2、整体 F 检验、预测和许多其他方面应当都未改变。最值得注意的差异在于对中变量的系数和标准误。

```
. regress csat Cpercent Cpercent2 high

      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  3,    47) =  193.37
       Model |  207225.103     3  69075.0343           Prob > F      =  0.0000
    Residual |   16789.407    47  357.221426           R-squared     =  0.9251
-------------+------------------------------           Adj R-squared =  0.9203
       Total |   224014.51    50   4480.2902           Root MSE      =   18.90

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    Cpercent |  -2.682362   .1119085   -23.97   0.000    -2.907493   -2.457231
   Cpercent2 |   .0536555   .0063678     8.43   0.000     .0408452    .0664659
        high |   2.986509   .4857502     6.15   0.000     2.009305    3.963712
       _cons |   680.2552   37.82329    17.99   0.000     604.1646    756.3458
------------------------------------------------------------------------------
```

在本例中，当 *Cpercent2* 被纳入到模型时，*Cpercent* 系数的标准误实际上更低(0.111 908 5 对比 0.161 621 93)。相应地，t 统计值更大。因此，看起来对中确实改善了系数估计值的精度。现在 VIF 表看起来问题不大了：与未对中回归时 *percent* 和 *percent2* 只有 4% 的独立变异相比，现在三个预测变量中的每一个都超过了 80%。

```
. vif

    Variable |       VIF       1/VIF
-------------+----------------------
    Cpercent |      1.20    0.831528
   Cpercent2 |      1.18    0.846991
        high |      1.03    0.969423
-------------+----------------------
    Mean VIF |      1.14
```

有时可用另一个检查多元共线性的诊断方法，就是检查估计系数（不是变量）之间的相关矩阵。要得到该矩阵，可在 **regress**、**anova** 或其他模型拟合程序之后键入：

```
. correlate, _coef

             | Cpercent Cperce~2     high    _cons
-------------+------------------------------------
    Cpercent |   1.0000
   Cpercent2 |  -0.3893   1.0000
        high |  -0.1700   0.1040   1.0000
       _cons |   0.2105  -0.2151  -0.9912   1.0000
```

系数对（pairs of coefficients）之间的高度相关表明，可能存在共线性问题。

通过加上选项 **covariance**，我们可以查看系数的方差协方差矩阵，标准误就是据此推出的：

```
. correlate, _coef covariance

             | Cpercent Cperce~2     high    _cons
-------------+------------------------------------
    Cpercent |  .012524
   Cpercent2 | -.000277  .000041
        high | -.009239  .000322  .235953
       _cons |  .891126 -.051817 -18.2105   1430.6
```

8 拟合曲线

基础的回归和相关方法都假定存在线性关系(linear relationships)。在有限的取值范围内,线性模型提供了对许多真实现象的合理而简单的近似。但是分析人员也许会碰到线性近似过于简单的情况,这就需要非线性(nonlinear)的替代办法。本章描述了三种针对非线性或曲线(curvilinear)关系进行建模的主要方法:

1.非参数方法(nonparametric methods),包括波段回归(band regression)和 lowess 修匀(lowess smoothing)。

2.转换变量(transformed variables)的线性回归("曲线回归"),包括 Box-Cox 方法。

3.非线性回归(nonlinear regression)。

非参数回归适宜作为一种探索性工具,因为它能在不要求分析人员事先设定某一特殊模型的情况下直观地概要描述数据的模式。转换变量将诸如 OLS 回归(**regress**)等线性参数模型的用途扩展到包含曲线关系的情况。但是,非线性回归则属于另一种不同类别的方法,它可以对内在非线性模型(intrinsically nonlinear models)的参数进行估计。

下列菜单选择涵盖了本章讨论的大部分操作。最后的非线性回归主题则需要采用命令方式来做。

Graphics-Twoway graph(scatterplot, line, etc.) 绘制两维标绘图

Statistics-Nonparametric analysis-Lowess smoothing 进行 lowess 修匀

Data-Create or change variables-Create new variable 进行变量转换

Statistics-Linear regression and related 进行线性回归及有关分析

命令示范

. boxcox *y x1 x2 x3*, model(lhs)

假定 $y^{(\lambda)}$ 是 *x1*、*x2* 和 *x3* 加上高斯等方差误差(*Gaussian constant-variance errors*)的线性函数的情况下,找到一种适合于对 y 进行 *Box-Cox* 转换的参数 λ(*lambda*)的最大似然估计值(*maximum-likelihood estimates*)。**model(lhs)** 选项限定是对左手边(left-hand-side)的变量 y 做转换。其他选项可以限定根据相同或不同的参数对右手边的变量(*x*)进行转换,也可以对模型的更多细节加以控制。请键入 **help boxcox** 查看命令语法和选项的完整清单。《基础参考手册》提供了技术细节。

```
. graph twoway mband y x, bands(10)  ||  scatter y x
```

画出 y 对 x 的散点图,并用线段将 10 个等宽垂直波段内的交叉中位数(cross-medians,即点(x 的中位数,y 的中位数))连接起来。这是"波段回归"的一种形式。键入 **mspline** 代替本命令中的 **mband** 将得到由修匀的立方样条曲线(smooth cubic spline curve)而不是线段连接起来的交叉中位数。

```
. graph twoway lowess y x, bwidth(.4)  ||  scatter y x
```

画出一条 lowess 修匀曲线(lowess-smoothed curve),并加上 y 对 x 的散点图。lowess 计算采用 0.4 波段宽度(bandwidth)(即 40% 的数据)。为了计算修匀的数值并将其作为新变量保存下来,请使用相关的命令 **lowess**。

```
. lowess y x, bwidth(.3) gen(newvar)
```

在 y 对 x 的散点图上采用 0.3 波段宽度(30% 的数据)画出一条 lowess 修匀曲线。这条曲线的预测值被存成一个名为 *newvar* 的变量。**lowess** 命令提供了比 **graph twoway lowess** 更多的选项,包括拟合方法和保存预测值的能力。详情请见 **help lowess**。

```
. nl exp2 y x
```

使用迭代非线性最小二乘法(iterative nonlinear least squares)拟合一个 2 参数的指数增长模型(exponential growth model),

预测值: $y = b_1 b_2^x$

其中的 **exp2** 指的是一个用于设定模型的单独程序。用户可以编一个程序来定义自己的模型,或者使用 Stata 提供的常见模型(包括指数、logistic 和 Gompertz 等模型)。在 **nl** 之后,使用 **predict** 能得到预测值或残差。

```
. nl log4 y x, init(B0=5, B1=25, B2=0.1, B3=50)
```

拟合一个 4 参数的 logistic 增长模型(**log4**),其形式为

预测值: $y = b_0 + b_1 / (1 + \exp(-b_2(x - b_3)))$

设定迭代估计过程的初始参数值为 $b_0 = 5$、$b_1 = 25$、$b_2 = 0.1$ 和 $b_3 = 50$。

```
. regress lny x1 sqrtx2 invx3
```

使用变量 lny、*x1*、*sqrtx2* 和 *invx3* 进行曲线回归。这些变量事先通过对原始变量 y、*x2* 和 *x3* 进行非线性转换得到,所用命令如下:

```
. generate lny = ln(y)
. generate sqrtx2 = sqrt(x2)
. generate invx3 = 1/x3
```

和本例中一样,当 y 变量被进行转换之后,由 **predict yhat** 得到的预测值或者由 **predict e, resid** 得到的残差也将具有被转换的单位(transformed units)。出于制图或其他目的,我们可能想将预测值或残差返回到原始数据单位上,请使用取逆转换(inverse transformations),比如:

```
. replace yhat = exp(yhat)
```

波段回归

非参数回归方法通常不形成一个明确的回归方程。它们基本上是展示 *y* 和 *x* 之间关系(可能是非线性的)的图形工具。Stata 可以在任何散点图或散点图矩阵之上画出波段回归这一简单的非参数回归类型。为了举例说明这点,考虑取自 MacKenzie (1990)的清醒认识冷战的数据(*missile.dta*)。观测案例是1958—1990 年期间由美国和(前)苏联在其军备竞赛过程中部署的 48 种远程核导弹:

```
Contains data from C:\data\missile.dta
  obs:            48                          Missiles (MacKenzie 1990)
 vars:             6                          16 Jul 2005 14:57
 size:         1 392 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
missile         str15  %15s                   Missile
country         byte   %8.0g       soviet     US or Soviet missile?
year            int    %8.0g                  Year of first deployment
type            byte   %8.0g       type       ICBM or submarine-launched?
range           int    %8.0g                  Range in nautical miles
CEP             float  %9.0g                  Circular Error Probable (miles)
-------------------------------------------------------------------------------
Sorted by:  country  year
```

missile.dta 中的变量包括一种被称作"圆概率误差"(Circular Error Probable,*CEP*)的准确性测量(accuracy measure)。*CEP* 代表一个圆圈的半径,中心是导弹弹着点,50% 的弹头都应当落在这个范围内。双方的科学家都致力于逐年改善导弹的准确性(图 8.1)。

```
. graph twoway mband CEP year, bands(8)
      ||  scatter CEP year
      ||  , ytitle("Circular Error Probable, miles") legend(off)
```

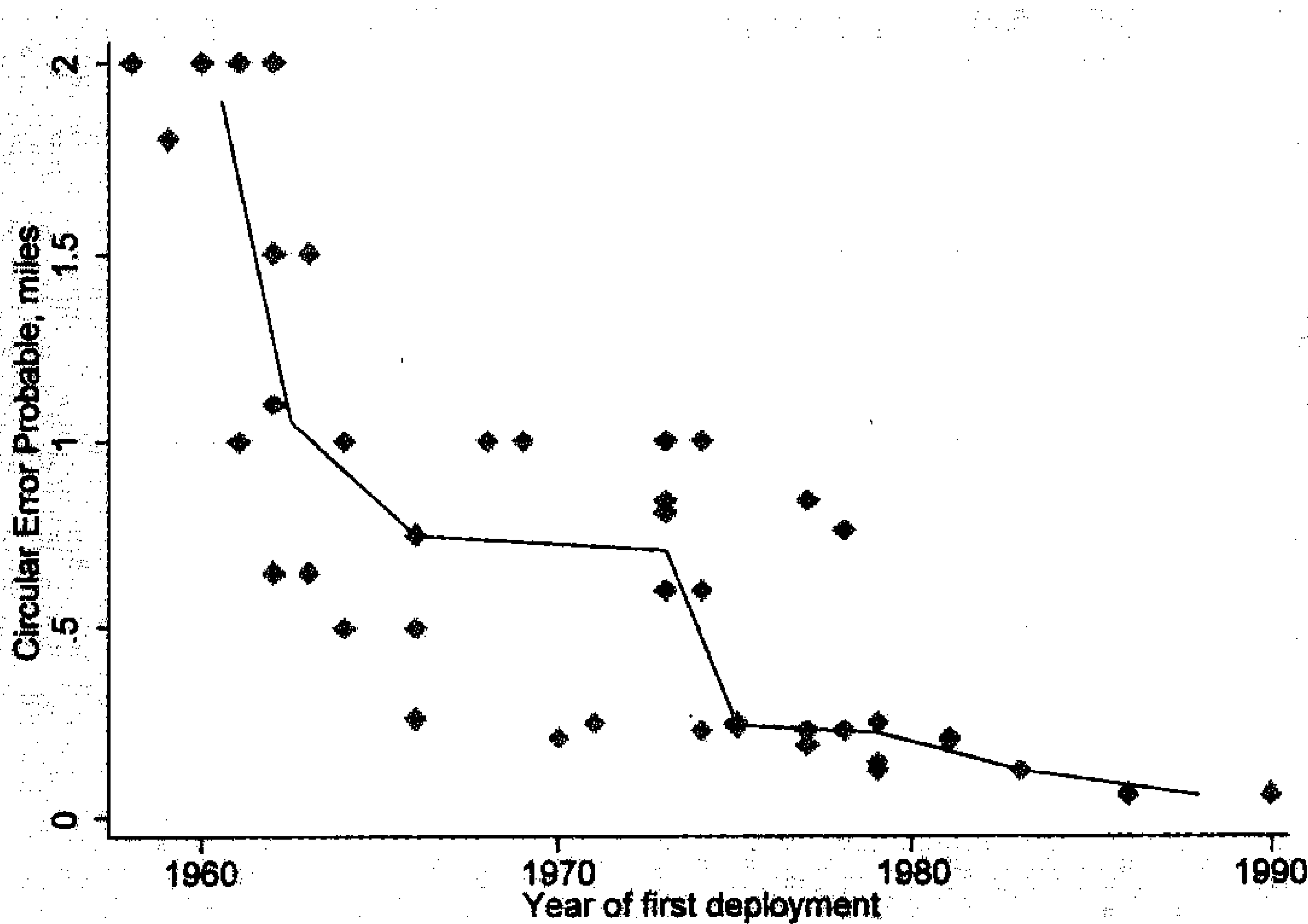


图 8.1

图 8.1 显示 *CEP* 随着时间推移而下降(准确性上升)。选项 **bands(8)** 设定 **graph twoway mband** 将散点图划分成 8 个等宽的垂直波段并使用线段将每一波段内的(*x* 中位数,*y* 中位数)点连接起来。这条曲线描绘了 *CEP* 的中位数如何随着年份 *year* 而变化。

非参数回归并不要求分析人员事先设定关系的函数形式。相反，它允许我们以“开放的心态”对数据进行探索。这一过程常常会揭示出一些令人感兴趣的结果，比如，当我们分别观察美国和(前)苏联导弹准确性的变化趋势时(图8.2)。下述命令中的 **by** (*country*)选项对每一个国家形成单独的标绘图，每幅图都将波段回归曲线和散点图叠并在一起。在 **by**( )选项内是控制图例(legend)和注释的子选项(suboptions)。

```
. graph twoway mband CEP year, bands(8)
      ||  scatter CEP year
      ||  , ytitle("Circular Error Probable, miles")
        by(country, legend(off) note(""))
```

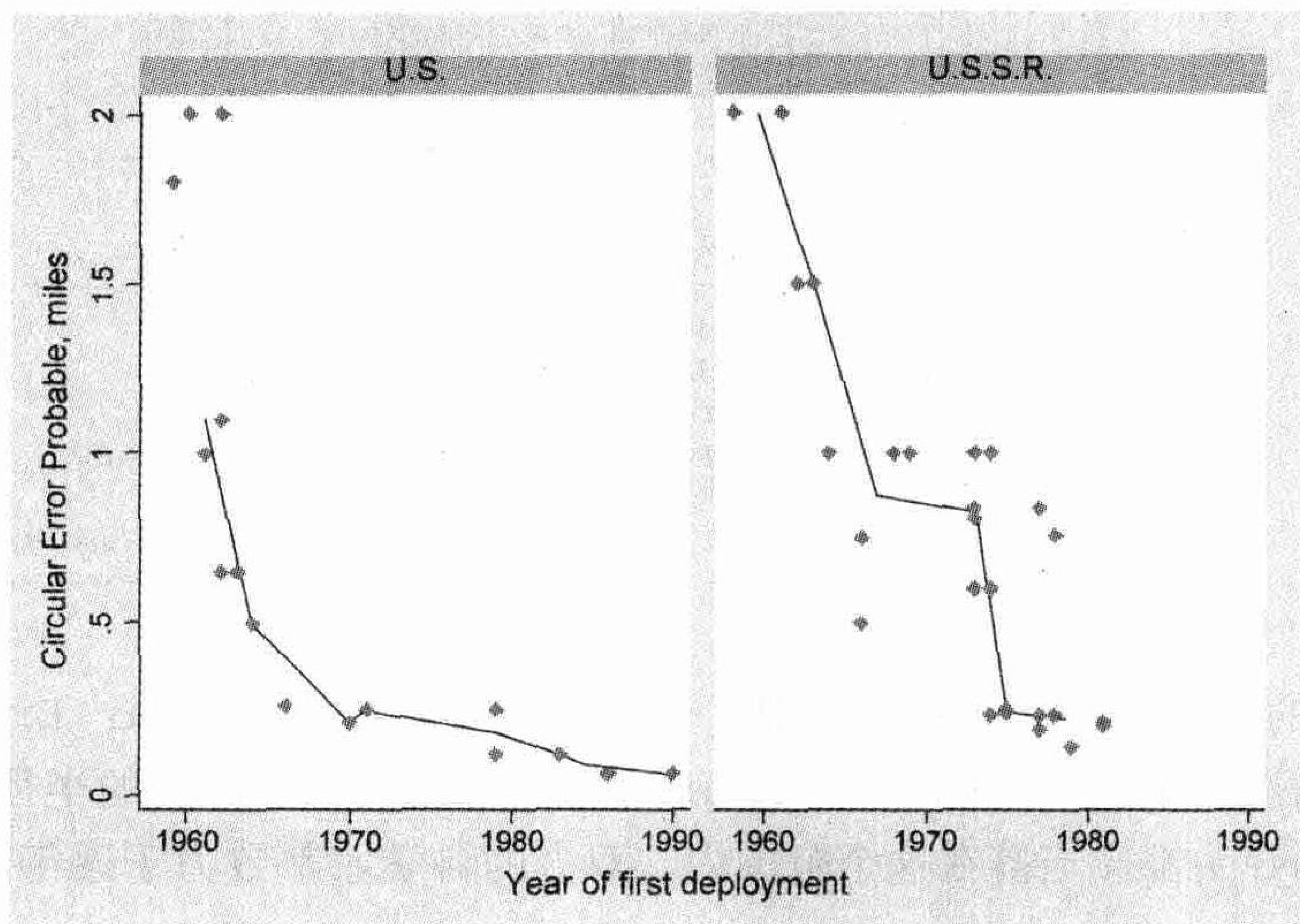


图8.2

图8.2中两条曲线的形状大为不同。美国的导弹在1960年代就已经比较准确，可以替换成更小的弹头。一个适合于先前只承载一个大弹头的同样大小的导弹可以装载三个或更多的小弹头。(前)苏联导弹的准确性改进得较慢，在1960年代晚期到1970年代初期这段时期内明显地出现停滞，因此落后于美国对手大约10年。为了弥补准确性上的不足，(前)苏联的策略侧重于装载高当量弹头的更大火箭。非参数回归能够辅助此类定性描述，或者作为拟合随后提到的那些参数模型的准备工作。

我们可以通过图形叠并将 **mband**(或 **mspline**)的波段回归曲线标绘图添加到任何散点图上。波段回归的简洁性使它成为一个便利的探索性工具，但是它还有一个值得注意的不足，即波段在 x 取值的不同区间都具有同样的宽度，尽管其中的一些波段包含很少的观测案例、甚至不包含观测案例。比如，对于服从正态分布的变量而言，数据密度(data density)在接近极值时会下降。因此，波段回归曲线的左端点或者右端点(它们往往决定着曲线的外观)常常只反映少量数据点。下一节将描述一种更为成熟、计算密集型的方法。

lowess 修匀

lowess 和 **graph twoway lowess** 可实现一种被称作 lowess 修匀(locally weighted scatterplot smoothing 的缩写，即局部加权散点图修匀)的非参数回归形式。由于具有可对拟合过程的细节进行控制的选项，**lowess** 命令总的来说更为专业

也更为强大。**graph twoway lowess** 具有简洁的优点，并遵循 **graph twoway** 这一族命令的习惯语法。下面的例子使用 **graph twoway lowess** 只对美国的导弹(*country*==0)画出 *CEP* 对 *year* 的标绘图。

如果我们改为键入以下命令，得到的图非常类似于图 8.2：

```
. lowess CEP year if country == 0, bwidth(.4)
```

和图 8.2 一样，图 8.3 显示美国导弹准确性在 1960 年代期间得到迅速改进，而在 1970 年代和 1980 年代则以更缓慢的速度发展着。这里得到了 *CEP* 的 lowess 修匀值，名为 *lsCEP*。选项 **bwidth(.4)** 设定 lowess 的波段宽度，即用于对每一点进行修匀的样本比例(fraction of sample)。默认设置是 **bwidth(.8)**。波段宽度越接近于 1，修匀的程度越高。

```
. graph twoway lowess CEP year if country == 0, bwidth(.4)
      || scatter CEP year
      || , legend(off) ytitle("Circular Error Probable, miles")
```

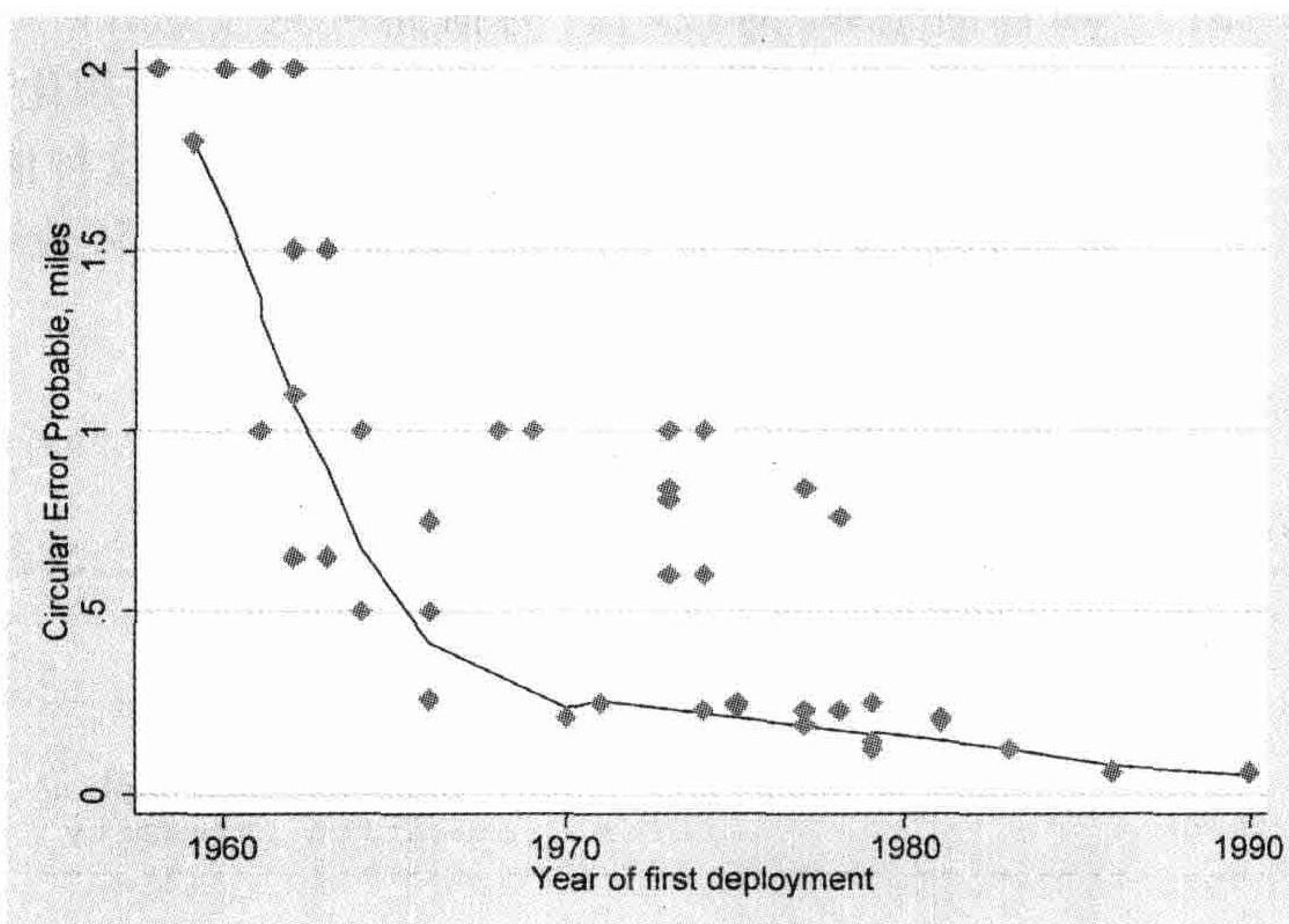


图 8.3

n 个观测案例的 lowess 预测(修匀) *y* 值由 *n* 个加权回归得到。设 *k* 表示波段总宽度的一半，并被平截成一个整数。对于每一 y_i，修匀值 y_i^s 通过只涉及那些位于从 $i = \max(1, i-k)$ 到 $i = \min(i+k, n)$ 区间内的观测案例的加权回归得到。根据一个局部加权函数(tricube)，这一区间内的第 *j* 个观测案例的权数为 w_j：

$$w_j = (1 - |u_j|^3)^3$$

这里，

$$u_j = (x_i - x_j)/\Delta$$

Δ 代表同一区间内 x_i 与其最远的观测案例之间的距离。当 $x_i = x_j$ 时，权数等于 1；但是在区间的边界处，权数下降到零。有关 lowess 方法的更详细的讨论和举例请见 Chambers 等(1983)或 Cleveland(1993)。

lowess 选项包括以下这些：

mean 用于移动平均数法修匀(running-mean smoothing)。默认选项为移动线最小二乘法修匀(running-line least squares smoothing)。

noweight 未加权的修匀(unweighted smoothing)。默认选项为 Cleveland 局部加权函数(tricube weighting function)。

bwidth( ) 设定波段宽度。除了接近端点处采用更小的非对中的波段之外,约有 bwidth × *n* 个观测案例的对中子集(centered subsets)被用于修匀。默认选项为 **bwidth(.8)**。

logit 将修匀值做 logit 转换。

adjust 调整修匀值的平均数,使其等于原始 y 变量的平均数;像 **logit** 一样,**adjust** 对二分类的 *y* 很有用。

gen(newvar) 创建包含 *y* 的修匀值的新变量。

nograph 抑止图形显示。

plot( ) 提供了一种将其他图形添加到已有图形中的办法;见 **help addplot_option**。

rplots( ) 改变参照线(reference line)的外观;参见 **help cline_options**。

由于要求进行 *n* 个加权回归,所以 lowess 修匀在大样本情况下运行较慢。

除了修匀散点图之外,**lowess** 也可用于探索性的时间序列修匀。文件 *ice.dta* 包含取自格陵兰冰原(GISP2)项目的结果,对该项目的描述可见于 Mayewski、Holdsworth 等(1993)和 Mayewski、Meeker 等(1993)。研究者们提取并用化学方法分析了代表超过 100 000 年气候史的冰芯。*ice.dta* 还包含了一小部分这种信息:测出的非海硫酸盐浓度(*sulfate*)和一个自公元 1500 年以来的“极地环流强度(Polar Circulation Intensity)”指数(*PCI*)。

```
Contains data from C:\data\ice.dta
  obs:           271                          Greenland ice (Mayewski 1995)
 vars:             3                          14 Jul 2005 14:57
 size:         5 962  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %ty                    Year
sulfate         double %10.0g                 SO4 ion concentration, ppb
PCI             double %6.0g                  Polar Circulation Intensity
-------------------------------------------------------------------------------
Sorted by:  year
```

为了保留取自这 271 个点的时间序列的更多详情,我们采用只占样本 5% 的相对狭窄的波段宽度进行修匀,图 8.4 画出了这一结果。修匀曲线已被画得更粗以便从视觉上将其与原始数据区分开来(有关线条宽度的其他选择,请键入 **help linewidthstyle**)。

主要来自火山或燃烧诸如煤和石油等化石燃料的非海硫酸盐(SO_4)在被排入到大气中之后到达了格陵兰冰原。不论是图 8.4 中的修匀曲线还是原始曲线都传递出这一信息。修匀曲线显示出从 1500 年到 1800 年代早期,平均数在摆动之中略有提升。1900 年以后,化石燃料促使修匀曲线明显提高,1929 年(大萧条)之后和 1970 年代早期(混杂着 1970 年美国清洁空气法案、1973 年阿拉伯石油禁运以及随后油价上涨等影响)出现了暂时性下降。原始数据中的大多数尖锋已被识别出是因为冰岛的 Hekla(1970)或阿拉斯加的 Katmai(1912)等世界著名火山的爆发。

```
. graph twoway lowess sulfate year, bwidth(.05) clwidth(thick)
     ||  line sulfate year, clpattern(solid)
     ||  , ytitle("SO4 ion concentration, ppb")
     legend(label(1 "lowess smoothed") label(2 "raw data"))
```

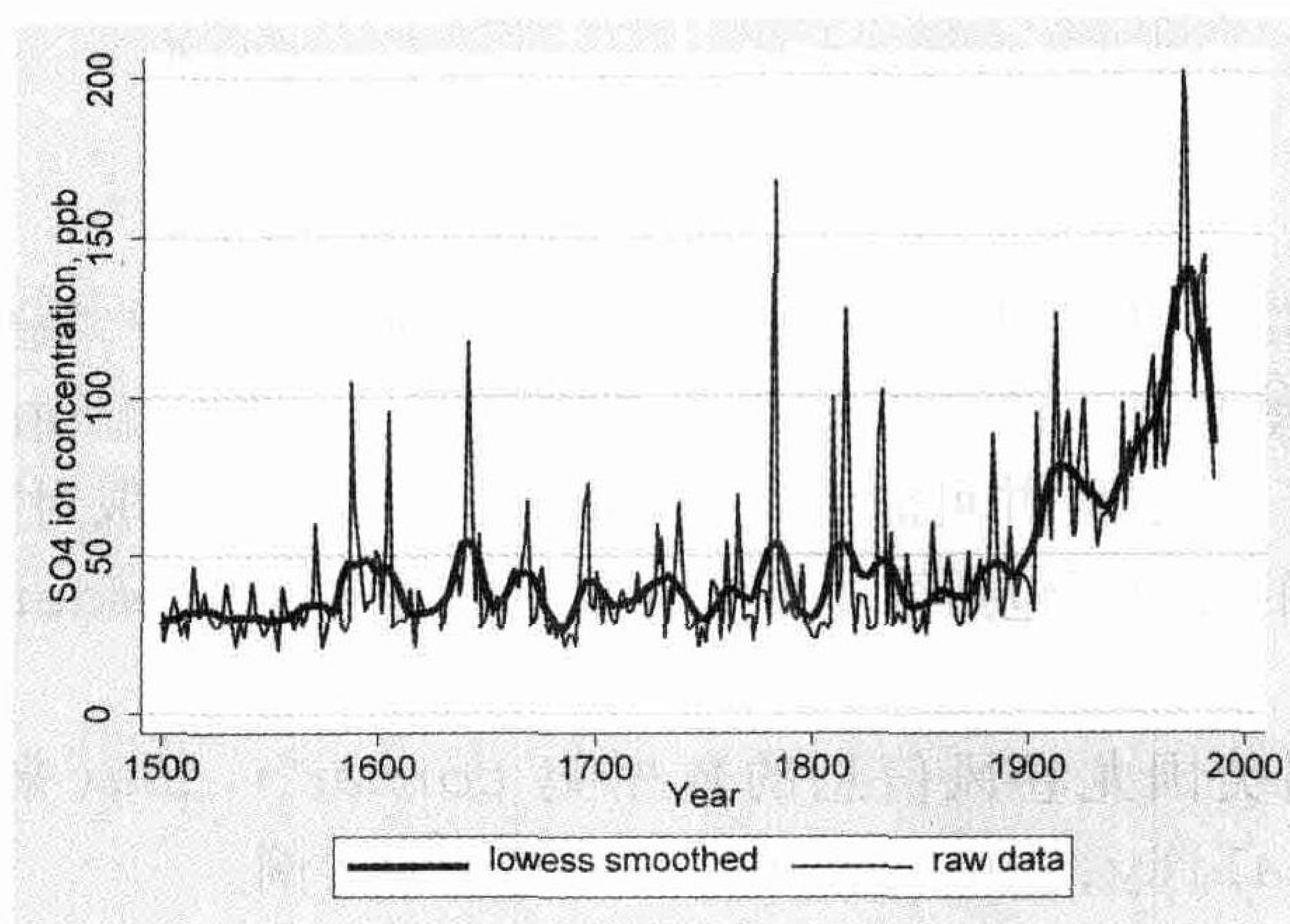


图 8.4

在对时间序列数据进行过修匀之后，分别对平滑序列和波动(残差)序列进行研究往往是很有用的。下述命令创建了两个新变量：lowess 修匀的硫酸盐取值(*smooth*)和原始数据减去修匀值计算出的残差或波动值(*rough*)。

```
. lowess sulfate year, bwidth(.05) gen(smooth)
. label variable smooth "SO4 ion concentration (smoothed)"
. gen rough = sulfate - smooth
. label variable rough "SO4 ion concentration (rough)"
```

通过 **text( )**选项标注图形然后加以合并，图 8.5 以成对标绘图的形式对 *smooth* 和 *rough* 两条时间序列进行了比较。

```
. graph twoway line smooth year, ylabel(0(50)150)  xtitle("")
     ytitle("Smoothed") text(20 1540 "Renaissance")
     text(20 1900 "Industrialization")
     text(90 1860 "Great Depression 1929")
     text(150 1935 "Oil Embargo 1973") saving(fig08_05a, replace)
. graph twoway line rough year, ylabel(0(50)150) xtitle("")
     ytitle("Rough") text(75 1630 "Awu 1640", orientation(vertical))
      text(120 1770 "Laki 1783", orientation(vertical))
     text(90 1805 "Tambora 1815", orientation(vertical))
     text(65 1902 "Katmai 1912", orientation(vertical))
     text(80 1960 "Hekla 1970", orientation(vertical))
     yline(0) saving(fig08_05b, replace)
. graph combine fig08_05a.gph fig08_05b.gph, rows(2)
```

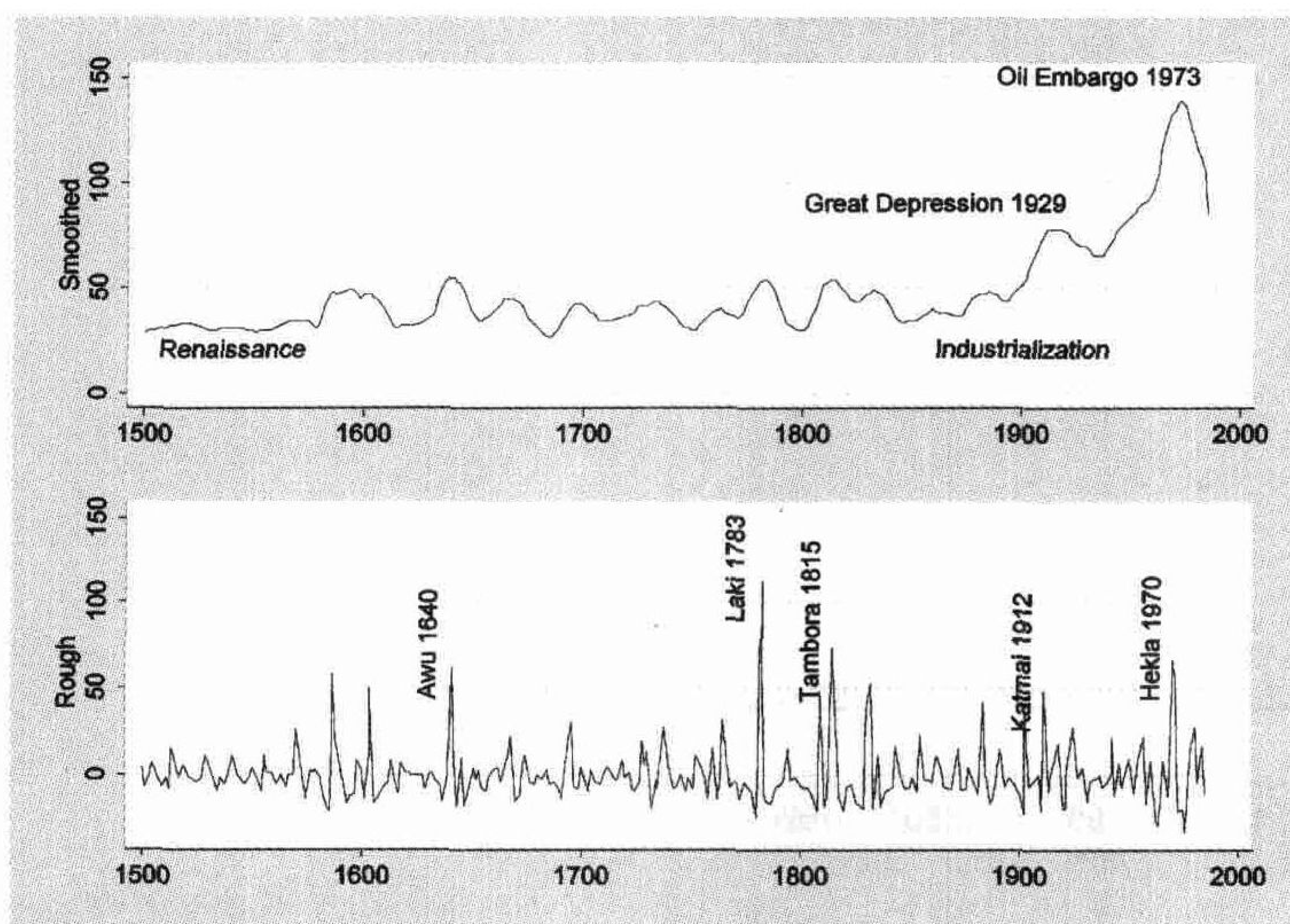


图 8.5

转换变量回归—1

通过对一个或更多变量作非线性转换，然后将转换变量纳入到线性回归中，我们就可以简单地对基础数据拟合一个曲线模型。第6章和第7章曾经给过一个这方面的多项式回归例子，即将预测变量 *x* 取了二次幂（并可能更高）同时作为预测变量。取对数也常常在很多领域中使用。其他常见的转换还包括第4章中介绍过的幂阶梯（ladder of powers）和 Box-Cox 转换。

包含1916年到1986年之间美国龙卷风信息的数据集 *tornado.dta*（来源：Council on Environmental Quality, 1988）提供了一个简单的实例。

```
Contains data from C:\data\tornado.dta
  obs:            71                          U.S. tornados 1916-1986
                                                (Council on Env. Quality 1988)
 vars:             4                          16 Jul 2005 14:57
 size:           994 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %8.0g                  Year
tornado         int    %8.0g                  Number of tornados
lives           int    %8.0g                  Number of lives lost
avlost          float  %9.0g                  Average lives lost/tornado
-------------------------------------------------------------------------------
Sorted by:  year
```

由于预报和监测龙卷风的能力不断提高，因而认定的龙卷风数量（*tornado*）也在增加，甚至包括那些危害很小的龙卷风，但是这一时期内由此丧生的数量（*lives*）在减少。因此，每年一场龙卷风的平均丧生数（*avlost*）随着时间推移在减少，但是线性回归（图8.6）未能很好地描述这一趋势。起初，散点下降得比回归线更快，然后在1950年代中期保持稳定。在后期，回归线竟然预测出后期的丧生人数为负值。此外，早期的平均丧生人数呈现出比晚期更大的变异，这表现出存在异方差性。

```
. graph twoway scatter avlost year
     || lfit avlost year, clpattern(solid)
     || , ytitle("Average number of lives lost") xlabel(1920(10)1990)
      xtitle("") legend(off) ylabel(0(1)7) yline(0)
```

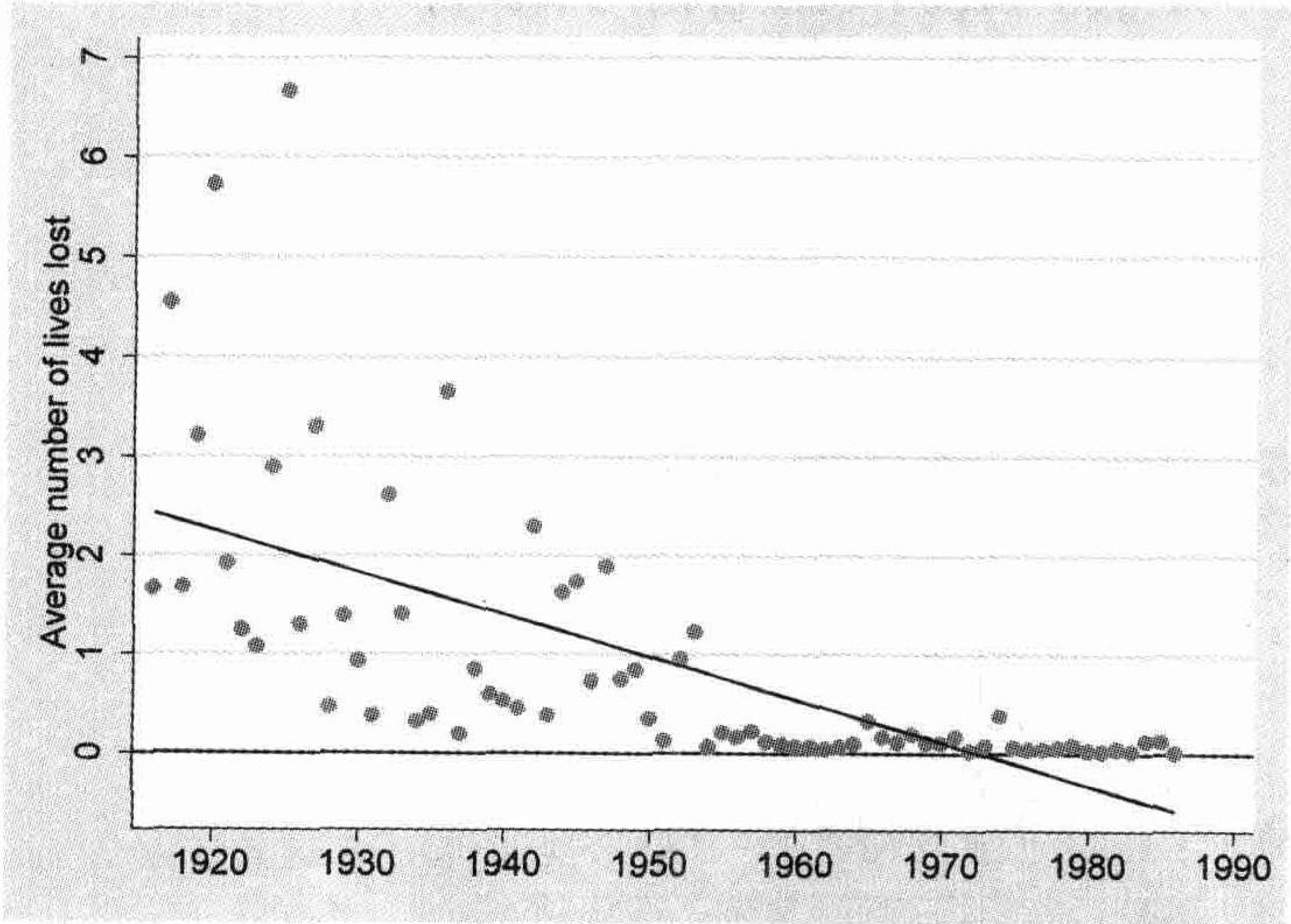


图8.6

如果我们换成对平均丧生人数的对数进行回归的话,关系就成了线性的,同时异方差性也会消失(图8.7)。

```
. generate loglost = ln(avlost)
. label variable loglost "ln(avlost)"
. regress loglost year

      Source |       SS       df       MS              Number of obs =      71
-------------+------------------------------           F(  1,    69) =  182.24
       Model |  115.895325     1  115.895325           Prob > F      =  0.0000
    Residual |  43.8807356    69  .63595269            R-squared     =  0.7254
-------------+------------------------------           Adj R-squared =  0.7214
       Total |   159.77606    70  2.28251515           Root MSE      =  .79747

------------------------------------------------------------------------------
    loglost |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       year |  -.0623418    .004618   -13.50   0.000    -.0715545   -.053129
      _cons |   120.5645   9.010312    13.38   0.000     102.5894    138.5395
------------------------------------------------------------------------------

. predict yhat2
(option xb assumed; fitted values)
. label variable yhat2 "ln(avlost) = 120.56 - .06year"
. label variable loglost "ln(avlost)"
. graph twoway scatter loglost year
     || mspline yhat2 year, clpattern(solid) bands(50)
     || , ytitle("Natural log(average lives lost)")
     xlabel(1920(10)1990) xtitle("") legend(off) ylabel(-4(1)2)
     yline(0)
```

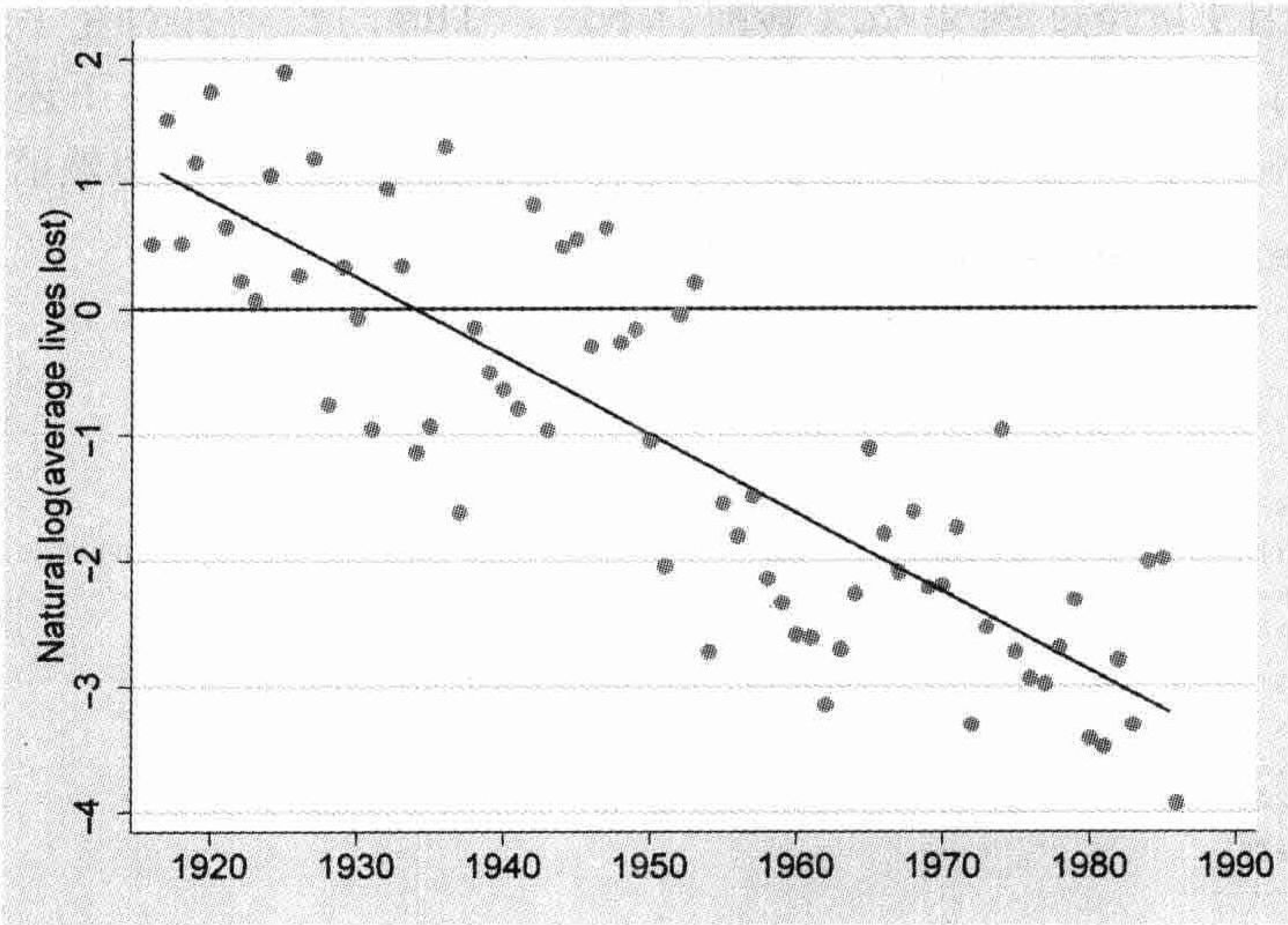


图 8.7

回归模型近似地为:

预测值: ln(*avlost*) = 120.56 -0.06*year*

因为我们是就丧生的对数对 *year* 进行回归,因此模型预测值也是以对数单位(logarithmic units)进行测量的。通过逆转换(inverse transformation)将这些预测值返回到其自然单位(丧生人数),这里就是对 *yhat2* 取 *e* 的指数幂:

```
. replace yhat2 = exp(yhat2)
(71 real changes made)
```

画出这些经逆转换的预测值，呈现出曲线回归模型（图8.8），该模型是我们通过对转换的 *y* 变量进行线性回归得到的。将图 *8.7* 和图 *8.8* 与图 *8.6* 进行比较会发现转换是如何使分析变得既简单又接近实际。

```
. graph twoway scatter avlost year
      || mspline yhat2 year, clpattern(solid) bands(50)
      || , ytitle("Average number of lives lost") xlabel(1920(10)1990)
       xtitle("") legend(off) ylabel(0(1)7) yline(0)
```

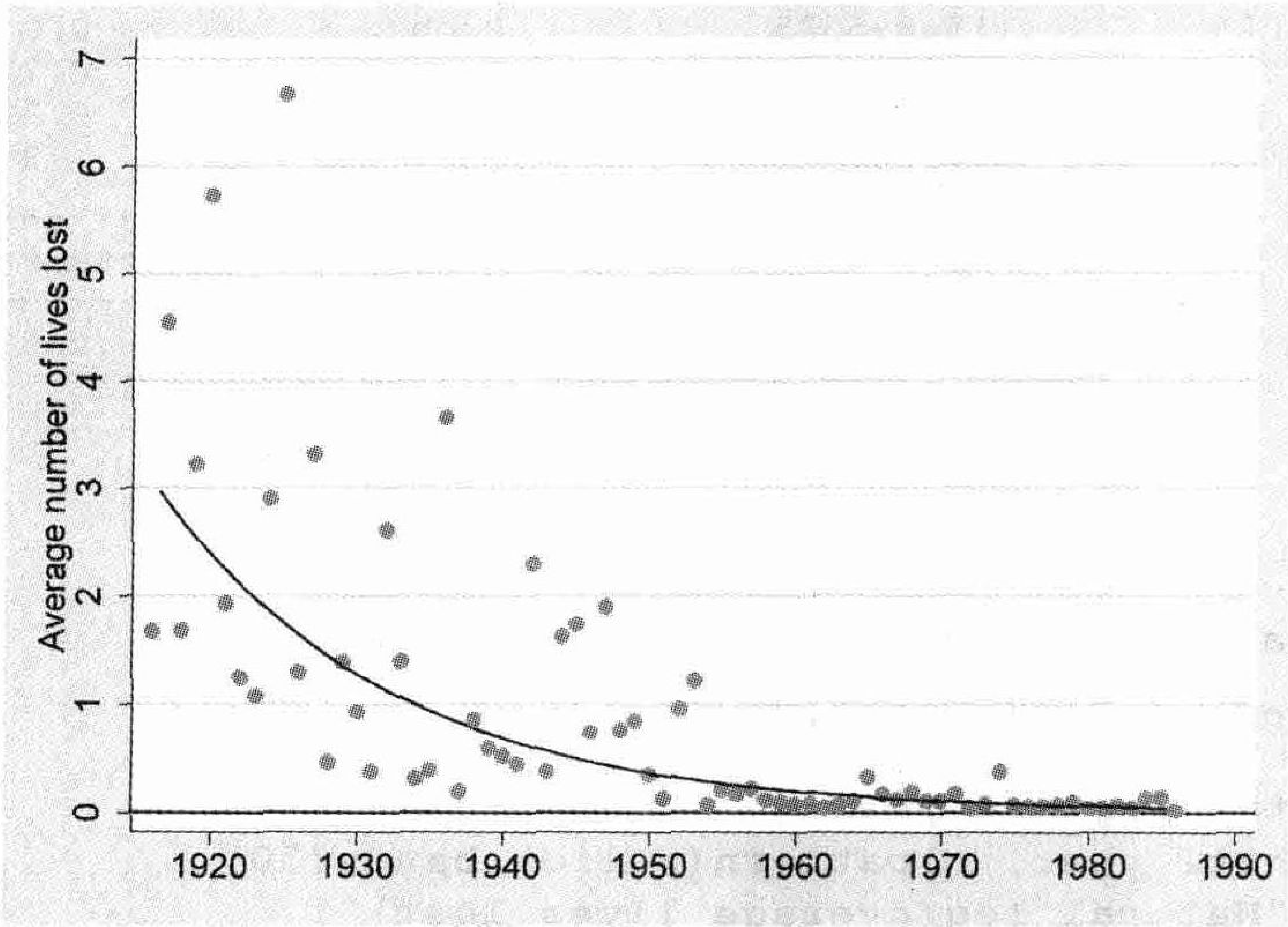


图 8.8

boxcox 命令使用最大似然方法来拟合经 Box-Cox 转换的曲线模型（第 4 章介绍过）。对龙卷风数据拟合因变量经过 Box-Cox 转换（**model(lhs)**）设定针对左手边的变量做转换）的模型，我们会得到非常类似于图8.7 和图8.8 对应模型的结果。下述命令中的 **nolog** 选项并不影响模型，只是要求不显示每次拟合迭代过程后的对数似然值。

```
. boxcox avlost year, model(lhs) nolog

                                                  Number of obs   =         71
                                                  LR chi2(1)      =      92.28
Log likelihood = -7.7185533                       Prob > chi2     =      0.000

------------------------------------------------------------------------------
      avlost |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /theta |  -.0560959   .0646726    -0.87   0.386    -.1828519     .07066
------------------------------------------------------------------------------

Estimates of scale-variant parameters
----------------------------
             |      Coef.
-------------+--------------
Notrans      |
        year |  -.0661891
       _cons |   127.9713
-------------+--------------
      /sigma |   .8301177
----------------------------

---------------------------------------------------------
   Test         Restricted      LR statistic      P-Value
    H0:       log likelihood        chi2        Prob > chi2
---------------------------------------------------------
theta = -1      -84.928791         154.42          0.000
theta =  0      -8.0941678           0.75          0.386
theta =  1      -101.50385         187.57          0.000
---------------------------------------------------------
```

boxcox 输出结果显示 theta = -0.056，作为对 *avlost* 进行转换的最优 Box-Cox 参数，以便将其与 *year* 之间的关系线性化。因此，左手边的转换为：

$$avlost^{(-0056)} = (avlost^{(-0.056)} - 1)/(-0.056)$$

以诸如-0 .056 这样一个接近零的参数进行 Box-Cox 转换得到的结果类似于我们前面“手工”对该变量所作的自然对数转换。因此，一点也不奇怪的是 **boxcox** 回归模型有预测值：

$$avlost^{(-0.056)} = 127.97 - 0.07year$$

类似于前面图 8.7 和图 8.8 中画出的模型（预测值为 ln(*avlost*) =120.56 - 0.06 *year*）。**boxcox** 程序假定误差为正态、独立和同分布。然而，它在选择转换方法时并不考虑使残差正态化。

boxcox 可以拟合几种不同的模型，包括以不同于 *y* 变量转换的参数对方程右边的一些或全部自变量进行转换的多元回归。它不能针对每个预测变量分别采用不同的转换。为了做到这点，我们在下一节将回过头来介绍一种“手工”曲线回归方法。

转换变量回归—2

对于多元回归的例子，我们将使用数据集 *nations.dta*（取自 World Bank, 1987;World Resources Institute, 1993）中有关109 个国家生活状况的数据。

```
Contains data from C:\data\nations.dta
  obs:          109                            Data on 109 nations, ca. 1985
 vars:           15                            16 Jul 2005 14:57
 size:        4 033  (99.9% of memory free)
---------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label       variable label
---------------------------------------------------------------------------
country         str8   %9s                     Country
pop             float  %9.0g                   1985 population in millions
birth           byte   %8.0g                   Crude birth rate/1000 people
death           byte   %8.0g                   Crude death rate/1000 people
chldmort        byte   %8.0g                   Child (1-4 yr) mortality 1985
infmort         int    %8.0g                   Infant (<1 yr) mortality 1985
life            byte   %8.0g                   Life expectancy at birth 1985
food            int    %8.0g                   Per capita daily calories 1985
energy          int    %8.0g                   Per cap energy consumed, kg oil
gnpcap          int    %8.0g                   Per capita GNP 1985
gnpgro          float  %9.0g                   Annual GNP growth % 65-85
urban           byte   %8.0g                   % population urban 1985

school1         int    %8.0g                   Primary enrollment % age-group
school2         byte   %8.0g                   Secondary enroll % age-group
school3         byte   %8.0g                   Higher ed. enroll % age-group
---------------------------------------------------------------------------
```

从图 8.9 中的散点图矩阵可以清楚地看到，出生率（*birth*）、人均国民生产总值（*gnpcap*）和儿童死亡率（*chldmort*）之间的关系并不是线性的。*gnpcap* 和 *chldmort* 的偏态分布也呈现出可能存在杠杆作用和类似影响问题。

```
. graph matrix gnpcap chldmort birth, half
```

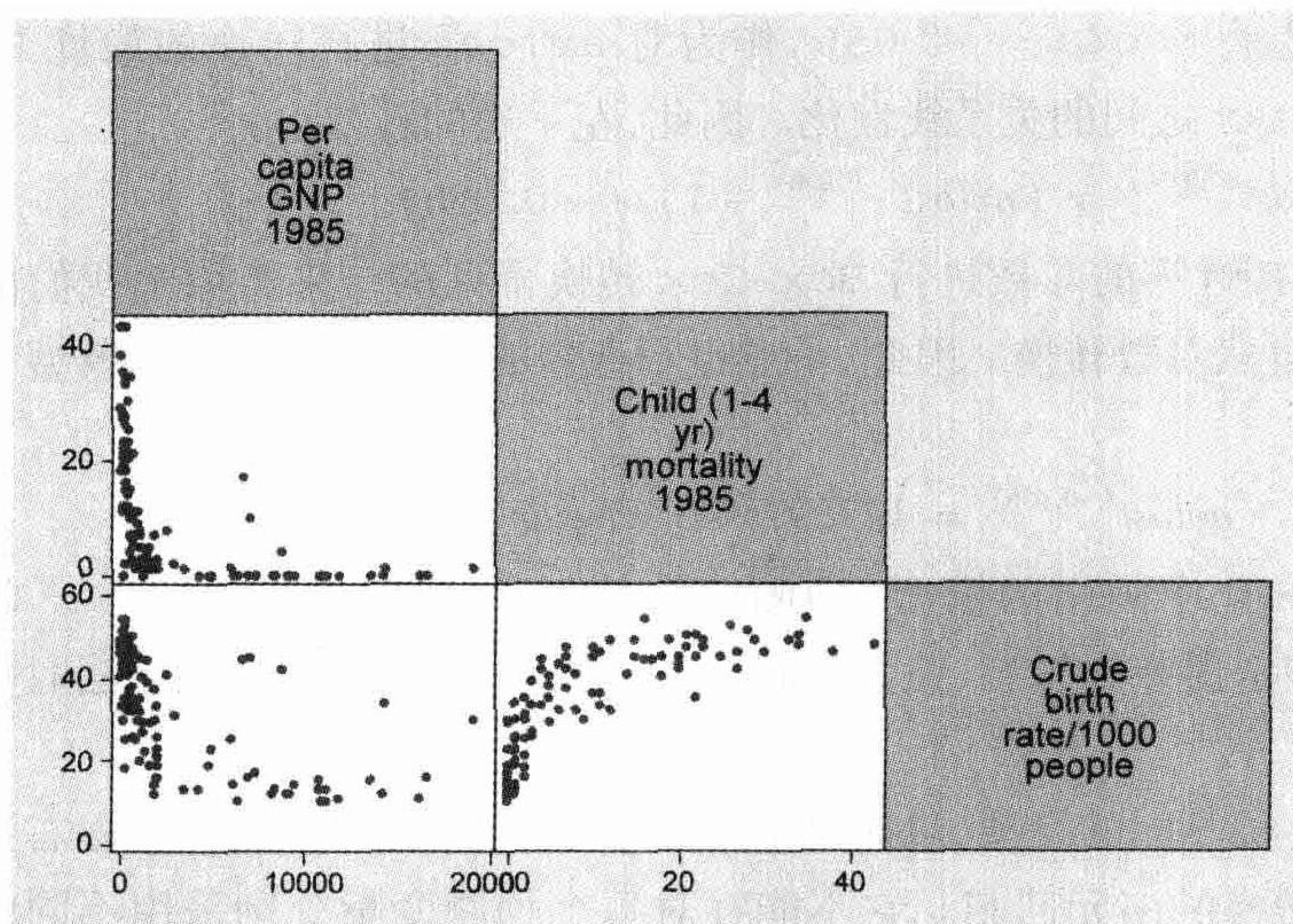


图 8.9

幂阶梯转换(ladder-of-powers transformations)的试验显示出,*gnpcap* 的对数和 *chldmort* 的平方根要比相应原始变量具有更为对称的分布、更少的特异值或潜在的杠杆作用点。更重要的是,这些转换大大消除了非线性关系,通过对比图 8.9 中的原始数据散点图和图 8.10 中其转换变量的散点图就能看到。

```
. generate loggnp = log10(gnpcap)
. label variable loggnp "Log-10 of per cap GNP"
. generate srmort = sqrt(chldmort)
. label variable srmort "Square root child mortality"
. graph matrix loggnp srmort birth, half
```

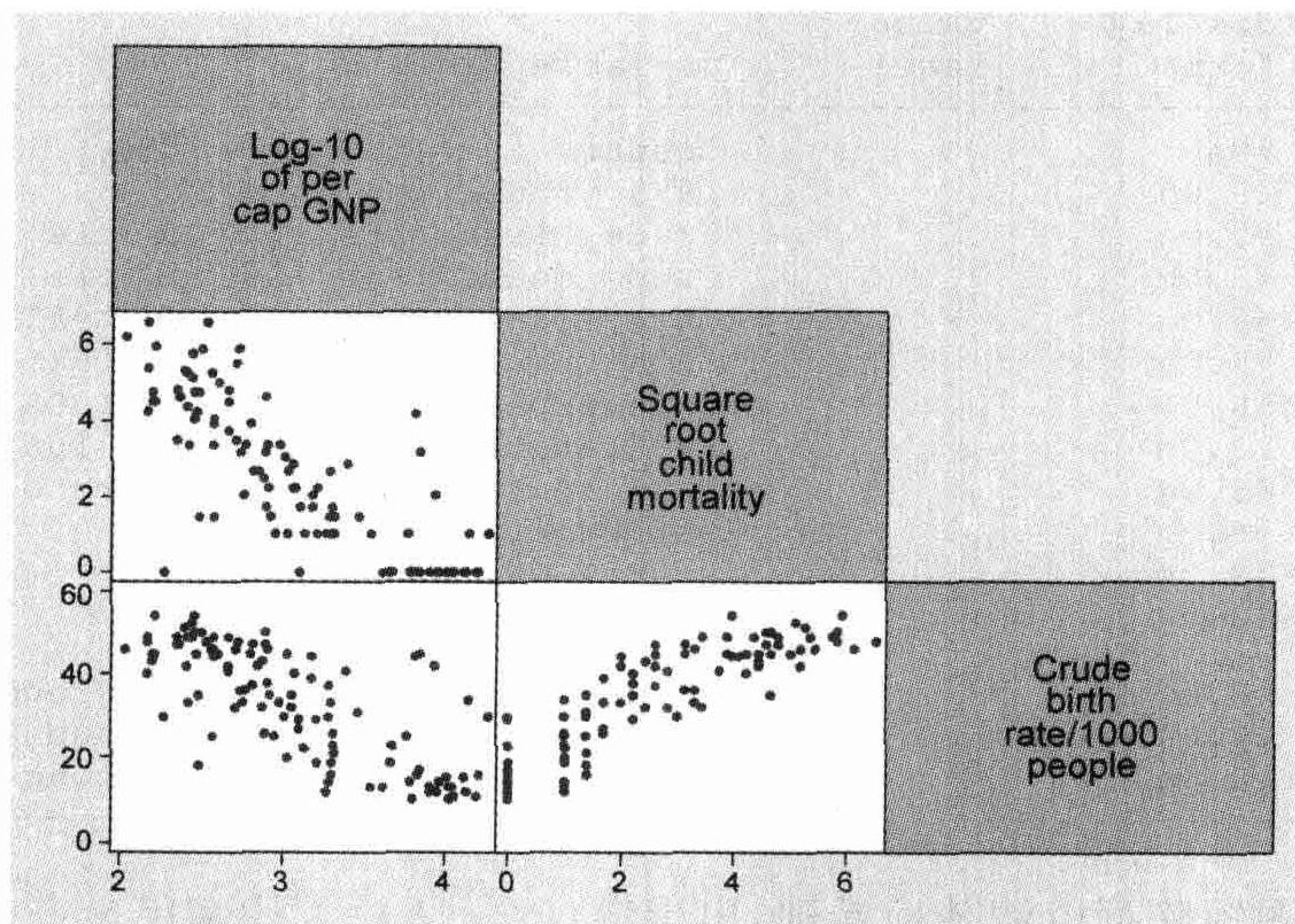


图 8.10

我们现在可以使用这些转换变量进行线性回归:

```
. regress birth loggnp srmort
```

```
      Source |       SS       df       MS              Number of obs =     109
-------------+------------------------------           F(  2,   106) =  198.06
       Model |  15837.9603     2  7918.98016           Prob > F      =  0.0000
    Residual |  4238.18646   106  39.9828911           R-squared     =  0.7889
-------------+------------------------------           Adj R-squared =  0.7849
       Total |  20076.1468   108  185.890248           Root MSE      =  6.3232

------------------------------------------------------------------------------
       birth |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      loggnp |  -2.353738   1.686255    -1.40   0.166    -5.696903    .9894259
      srmort |   5.577359    .533567    10.45   0.000     4.51951     6.635207
       _cons |   26.19488   6.362687     4.12   0.000     13.58024    38.80953
------------------------------------------------------------------------------
```

不同于原始数据的回归(未显示),这一转换变量的回归表明:一旦我们对儿童死亡率加以控制,人均国民生产总值对出生率的影响并不显著。转换变量回归拟合得略微更好一些:R_a^2 =0.784 9 而不是 0.671 5。(我们这里之可以对跨模型的 R_a^2 进行比较只是因为两个模型都有相同的未转换的 *y* 变量。)杠杆作用标绘图将证实,转换已经大大削减了原始数据回归中的曲线关系。

条件效应标绘图

条件效应标绘图(conditional effect plots)描绘了在其他 *x* 变量保持在诸如平均数、中位数、四分位数或极端值等任意数值上不变时按某一 *x* 变量函数的 *y* 的预测值。此类标绘图有助于对转换变量回归结果进行解释。

继续前面的例子,我们可以计算在 *srmort* 保持在其平均数(2.49)不变时按 *loggnp* 的函数的出生率预测值:

```
. generate yhat1 = _b[_cons] + _b[loggnp]*loggnp + _b[srmort]*2.49
. label variable yhat1 "birth = f(gnpcap | srmort = 2.49)
```

命令中的 _b[*varname*]这一项指的是最近一次回归的系数 *varname* 的回归系数。而 _b[_cons]为 *y* 的截距或常数。

为了得到一个条件效应标绘图,画出 *yhat1*(尽管此处不需要,但是如果需要可进行逆转换)对转换的 x 变量的图形(图 8.11)。因为条件效应标绘图不显示数据散点,因此添加诸如位于 *x* 变量的第 10 和第 90 百分位点这样的参照线(reference lines)可能会很有用,如图 8.11 所示。

```
. graph twoway line yhat1 gnpcap, sort xlabel(,grid) xline(230 10890)
```

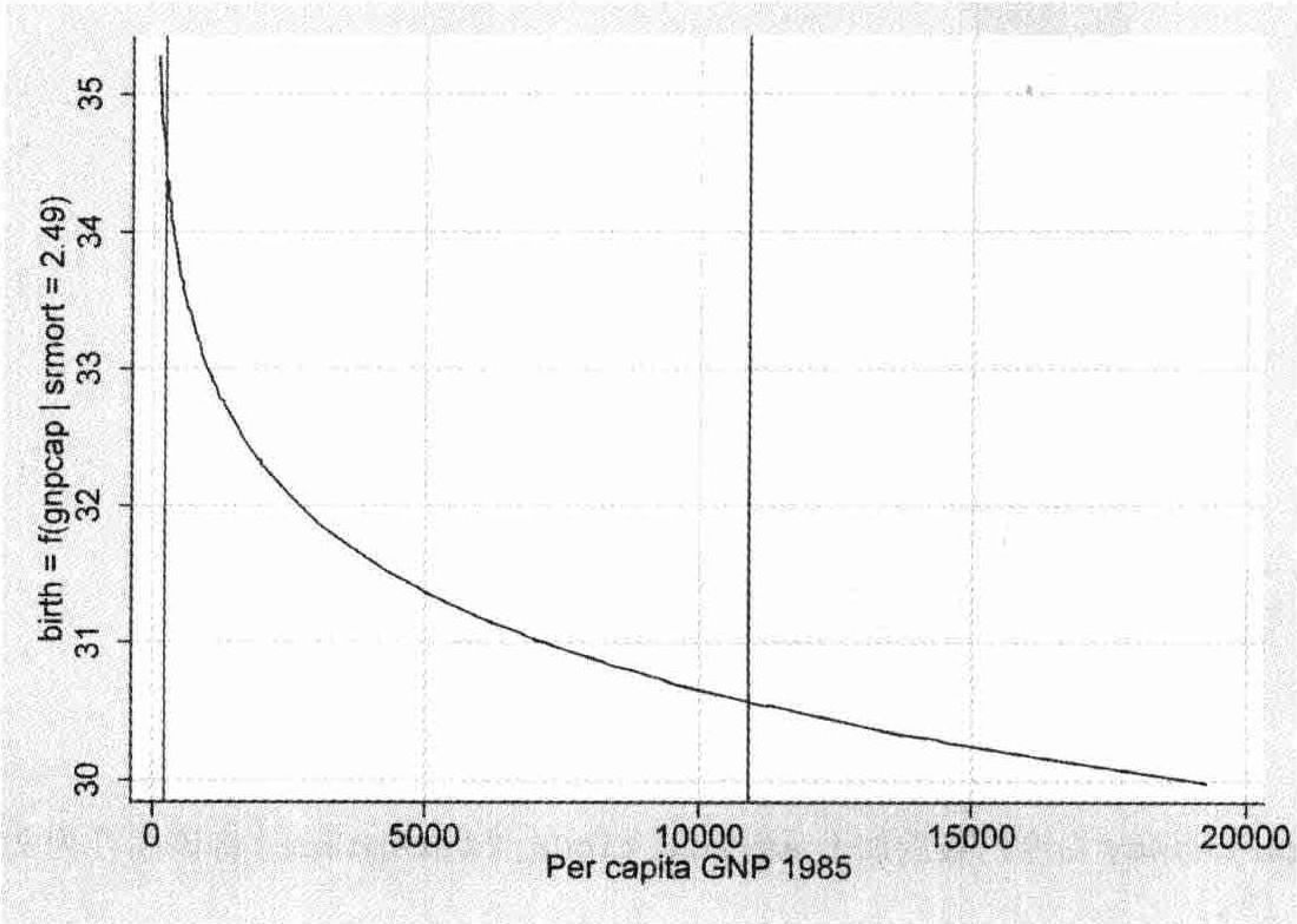


图 8.11

类似地,图 8.12 描绘了在 *loggnp* 保持在其平均数(3.09)时按 *srmort* 的函数的预测出生率:

```
. generate yhat2 = _b[_cons] + _b[loggnp]*3.09 + _b[srmort]*srmort
. label variable yhat2 "birth = f(chldmort | loggnp = 3.09)"
. graph twoway line yhat2 chldmort, sort xlabel(,grid) xline(0 27)
```

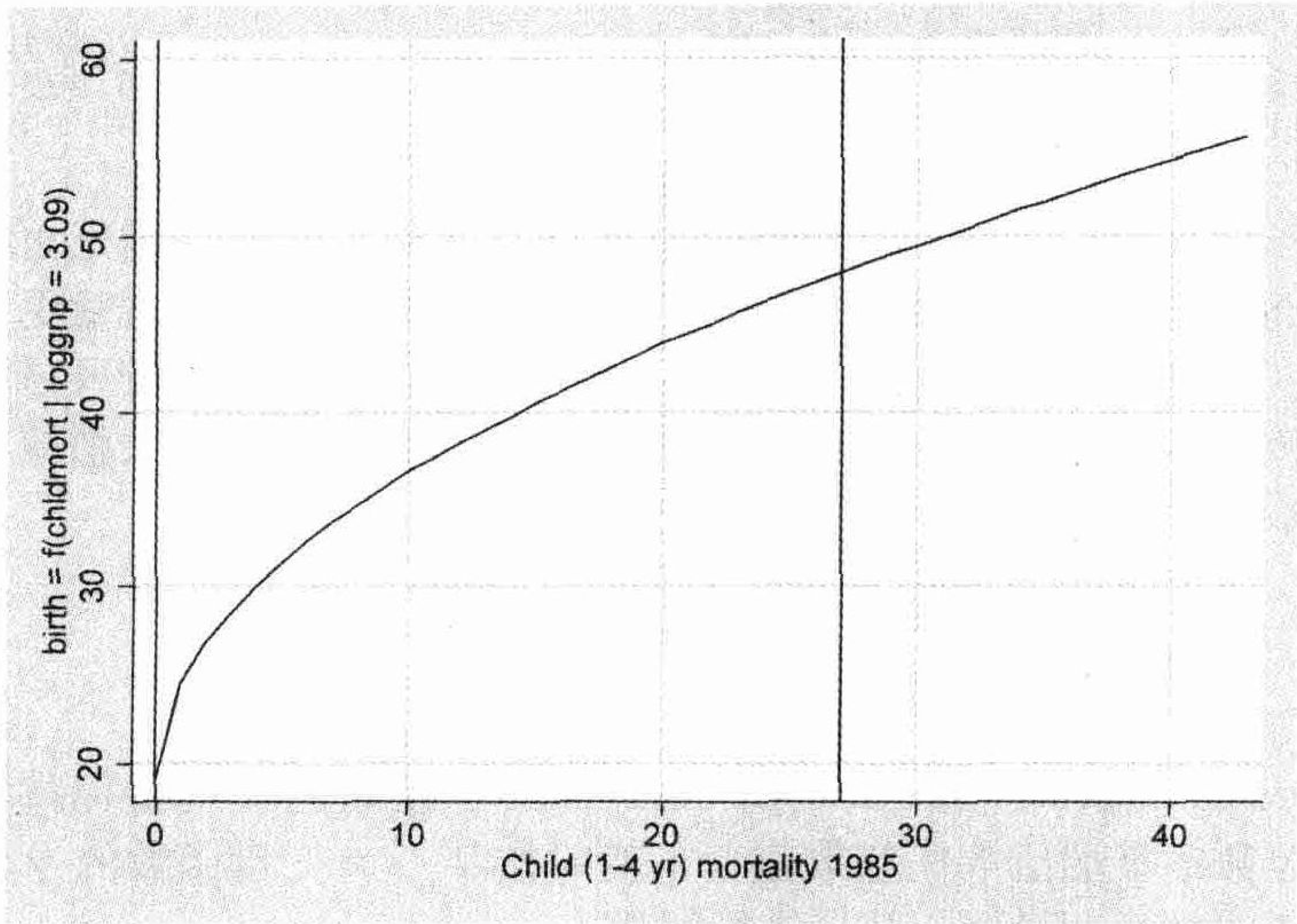


图 8.12

我们如何才能对不同 x 变量效应的强度进行比较呢? 标准化回归系数(beta 权数)有时候被用于这一目的,但是它们所采用的专门化"强度"定义又很容易令人误解。一种更有实际意义的比较可能来自于查看基于相同 y 的刻度画成的条件效应标绘图,这可以通过使用 **graph combine** 并设定共同的 y 轴刻度来轻松实现,就像图 8.13 那样,预测值曲线经过的垂直距离,尤其是 x 取值中间 80% 的区间内(位于第 10 和第 90 百分位点处的两条线之间)的距离,提供了一种效应幅度的直观比较[9]。

```
. graph combine fig08_11.gph fig08_12.gph, ycommon cols(2) scale(1.25)
```

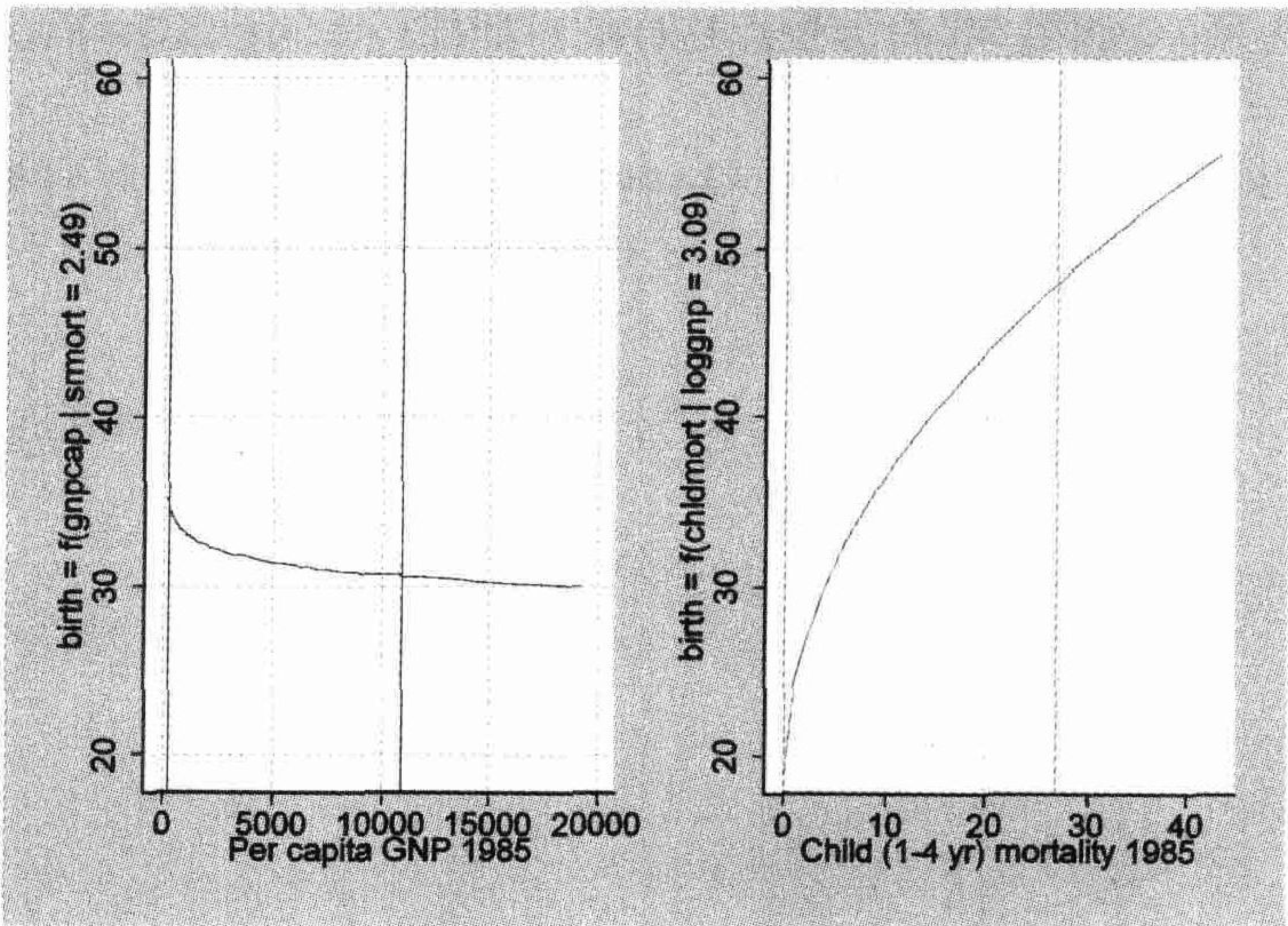


图 8.13

[9]【译注:可在画图 8.11 的 **graph twoway** 命令的最后加上 **saving(*fig08_11*,replace)** 将该图存为文件,图 8.12 也这样做,便可画出图 8.13。】

如图 8.13 中那样,将几个条件效应标绘图合并成一幅具有共同纵轴刻度的图像,可以对不同效应的强度进行迅速直观的比较。图 8.13 清楚地反映出儿童死亡率对出生率的影响要强大得多,而单独的标绘图(图 8.11 和图 8.12)却做不到这点。

非线性回归—1

变量转换通过使用熟悉的内在(intrinsically)线性模型技术使得拟合一些曲线关系成为可能。但是,内在非线性模型需要不同类型的拟合技术。命令 **nl** 使用迭代最小二乘法(iterative least squares)进行非线性回归(nonlinear regression)。本节使用一个简单例子的数据集 *nonlin.dta* 对此加以介绍:

```
Contains data from C:\data\nonlin.dta
  obs:          100                           Nonlinear model examples
                                                (artificial data)
 vars:            5                           16 Jul 2005 14:57
 size:        2 100 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
x               byte   %9.0g                  Independent variable
y1              float  %9.0g                  y1 = 10 * 1.03^x + e
y2              float  %9.0g                  y2 = 10 * (1 - .95^x) + e
y3              float  %9.0g                  y3 = 5 + 25/(1+exp(-.1*(x-50)))
                                                + e
y4              float  %9.0g                  y4 = 5 +
                                                25*exp(-exp(-.1*(x-50))) + e
-------------------------------------------------------------------------------
Sorted by:  x
```

nonlin.dta 数据是人工构造的,其中 y 变量被定义成 x 的多种非线性函数加上随机高斯误差(random Gaussian errors)。比如,*y1* 表示指数增长过程 $y1 = 10 \times 1.03^x$。根据该数据估计这些参数,**nl** 得到 $y1 = 11.20 \times 1.03^x$,它相当接近真实模型。

```
. nl exp2 y1 x
(obs = 100)

Iteration 0:  residual SS =  27625.96
Iteration 1:  residual SS =  26547.42
Iteration 2:  residual SS =   26138.3
Iteration 3:  residual SS =  26138.29

      Source |       SS       df       MS            Number of obs =       100
-------------+------------------------------         F(  2,     98) =   1250.42
       Model |  667018.255     2  333509.128         Prob > F      =    0.0000
    Residual |  26138.2933    98  266.717278         R-squared     =    0.9623
-------------+------------------------------         Adj R-squared =    0.9615
       Total |  693156.549   100  6931.56549         Root MSE      =  16.33148
                                                     Res. dev.     =  840.3864

2-param. exp. growth curve, y1=b1*b2^x
------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          b1 |   11.20416   1.146682     9.77   0.000     8.928602    13.47971
          b2 |   1.028838   .0012404   829.41   0.000     1.026376    1.031299
------------------------------------------------------------------------------
  (SE's, P values, CI's, and correlations are asymptotic approximations)
```

predict 命令获得由 **nl** 估计得到的非线性模型的预测值和残差。图 8.14 画出了前例中的预测值,表明模型和数据之间紧密拟合($R^2 = 0.96$)。

```
. predict yhat1
(option yhat assumed; fitted values)
. graph twoway scatter y1 x
      ||  line yhat1 x, sort
      ||  , legend(off) ytitle("y1 = 10 * 1.03^x + e") xtitle("x")
```

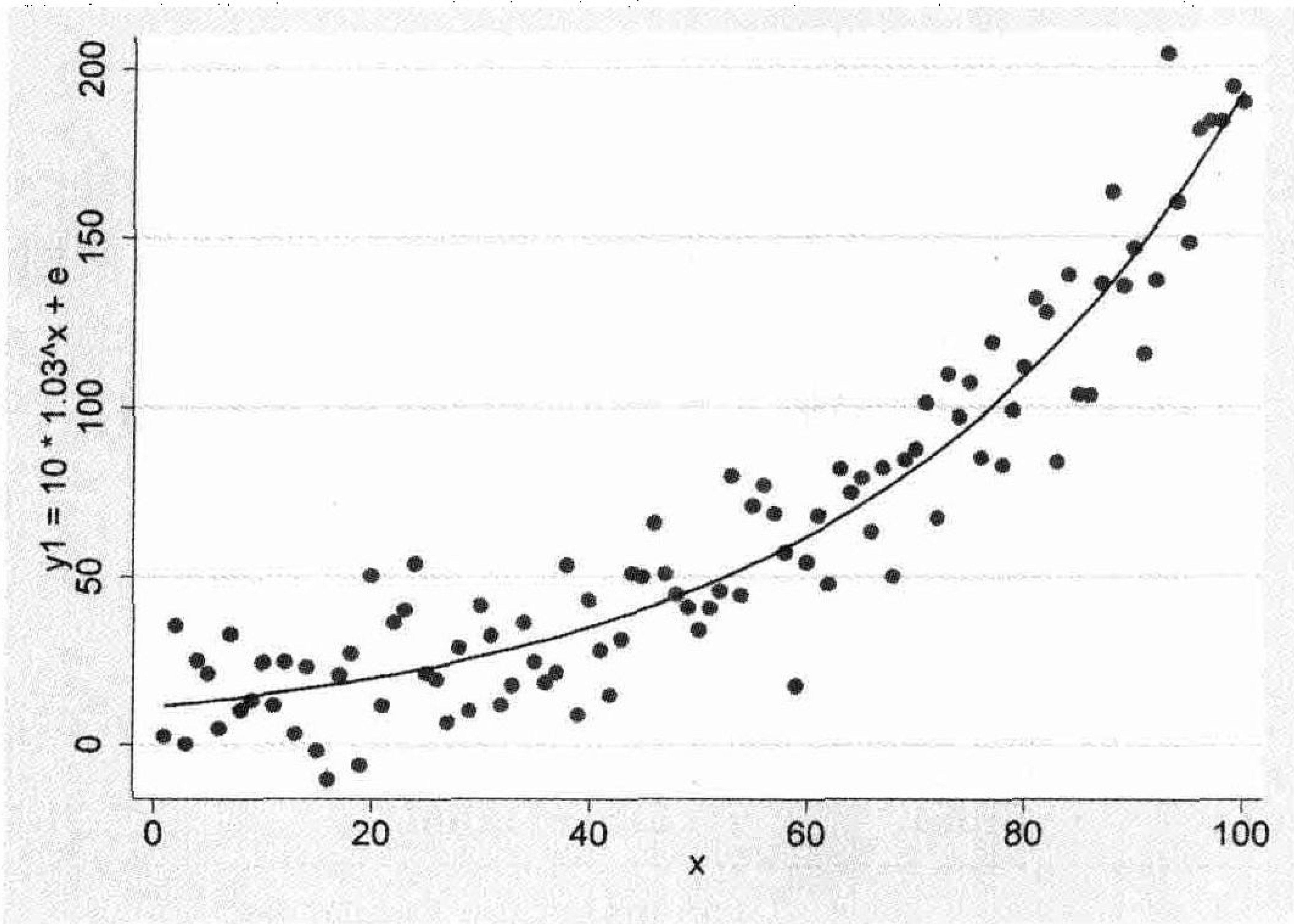


图 8.14

在命令 **nl exp2 y1 x** 中的 **exp2** 部分是调用名为 *nlexp2.ado* 的简单程序来设定一种特定的指数增长函数。Stata 提供了数个此类程序，定义了下列函数：

exp3 3 参数的指数函数：$y = b_0 + b_1 b_2^{x}$

exp2 2 参数的指数函数：$y = b_1 b_2^{x}$

exp2a 2 参数的负指数函数：$y = b_1(1 - b_2^{x})$

log4 4 参数的 logistic 函数，b_0 为初始水平，$(b_0 + b_1)$为渐近上限：

$$y = b_0 + b_1 / (1 + \exp(-b_2(x - b_3)))$$

log3 3 参数的 logistic 函数，0 为初始水平，b_1 为渐近上限：

$$y = b_1 / (1 + \exp(-b_2(x - b_3)))$$

gom4 4 参数的 Gompertz 函数，b_0 为初始水平，$b_0 + b_1$ 为渐近上限：

$$y = b_0 + b_1 \exp(-\exp(-b_2(x - b_3)))$$

gom3 3 参数的 Gompertz 函数，0 为初始水平，b_1 为渐近上限：

$$y = b_1 \exp(-\exp(-b_2(x - b_3)))$$

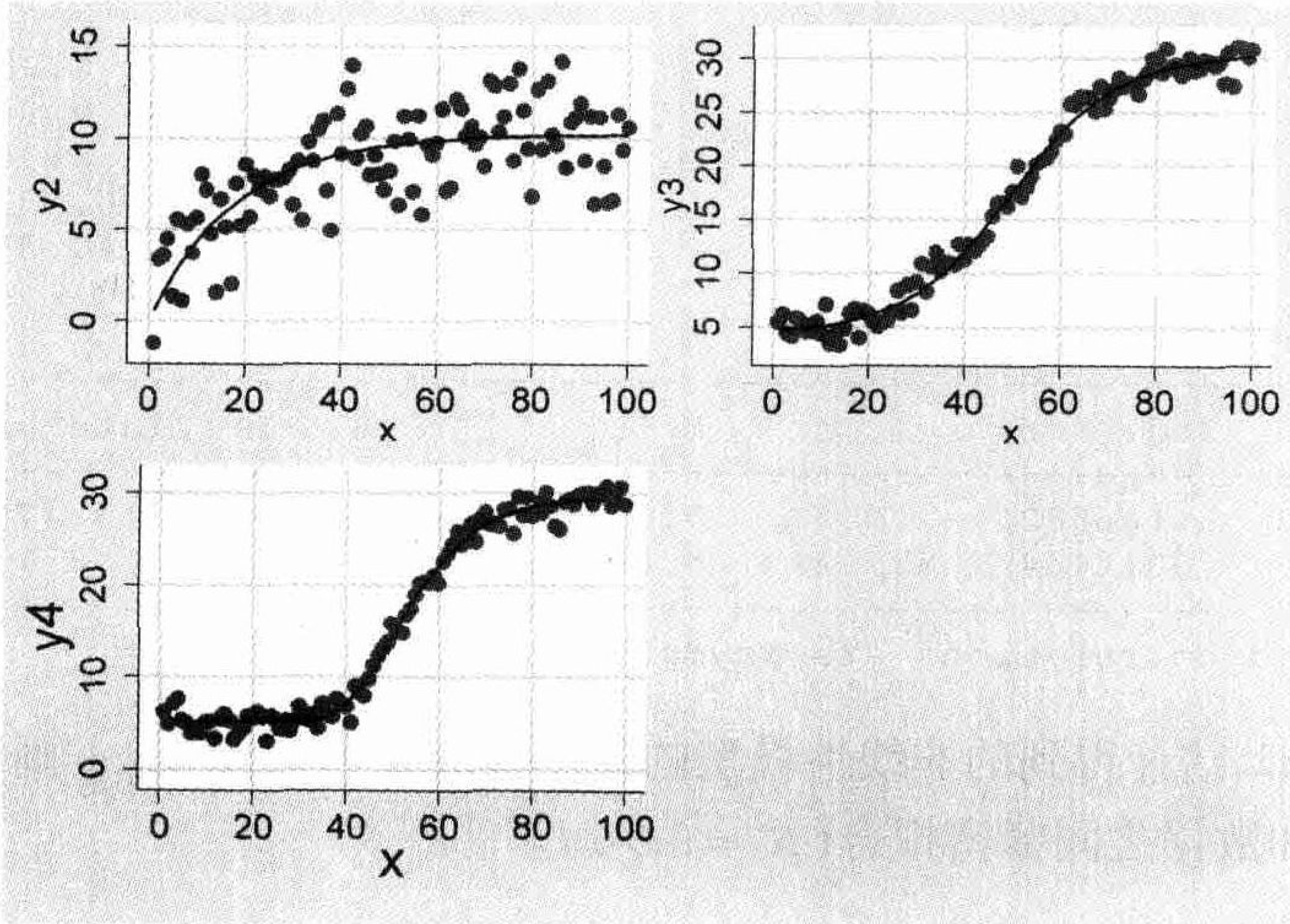


图 8.15

nonlin.dta 包含了与 **exp2**（*y1*）、**exp2 a**（*y2*）、**log4**（*y3*）和 **gom4**（*y4*）函数对应的例子。图 8.15 显示了使用 **nl** 对 *y2*、*y3* 和 *y4* 进行拟合得到的曲线。

用户也可以进一步编写自己的 **nl** 函数(nl*function*)程序。这是用于定义一个 2 参数指数增长模型的 nlexp2.ado 程序的代码：

```
*! version 1.1.3  12jun1998
program define nlexp2
    version 6
    if "`1'"=="?" {
        global S_2 "2-param. exp. growth curve, $S_E_depv=b1*b2^`2'"
        global S_1 "b1 b2"
/*
    Approximate initial values by regression of log Y on X.
*/
        local exp "[`e(wtype)' `e(wexp)']"
        tempvar Y
        quietly {
            gen `Y' = log(`e(depvar)') if e(sample)
            reg `Y' `2' `exp' if e(sample)
        }
        global b1 = exp(_b[_cons])
        global b2 = exp(_b[`2'])
        exit
    }
    replace `1'=$b1*($b2)^`2'
end
```

这个程序为待定参数找到一些近似的初始值(initial values)，将这些值存成名为 b1 和 b2 的“全局宏”(global macros)。然后应用初始参数估计值和以下模型方程计算出一套初始的预测值，作为一个名为 1 的“局部宏”(local macro)：

replace `1' = $b1 *($b2)^`2'

随后的 **nl** 迭代将返回到这一行，计算出新的预测值(替换宏 1 中的内容)作为它们对参数估计值 b1 和 b2 的改进。在 Stata 程序中，符号$ b1 意指“全局宏 b1 的内容”。类似地，符号 `1' 意指“局部宏 1 的内容”。

在用户尝试编写自己的非线性函数时，请研究 nllog4.ado、nlgom4.ado 和其他示例程序，同时参考有关手册或键入 **help nl** 寻找说明。第 14 章还有 Stata 的宏和其他编程方面的进一步讨论。

非线性回归—2

我们的第二个例子涉及真实数据，并且将示范研究中有帮助的一些步骤。数据集 *lichen.dta* 涉及对挪威极地斯瓦尔巴群岛苔藓(lichen)生长的测量(取自 Werner, 1990)。这些生长缓慢的共生体常常被用来断定岩石遗迹和其他沉积物的年代，因此它们的增长率(growth rates)令数个领域的科学家感兴趣。

正如图 8.16 中的 lowess 修匀曲线显示的那样，苔藓典型地呈现出早期生长相对较快，然后逐渐放慢的生长过程阶段性。

苔藓计量学家通过画出增长曲线(growth curve)来总结和比较此类模式。这些增长曲线可能并没有使用明确的数学模型，但是我们在这里可以拟合一条曲线来举例说明非线性回归的过程。Gompertz 曲线是已被广泛用于对生物生长过程进行建模的非对称的 S 形曲线：

```
Contains data from C:\data\lichen.dta
  obs:            11                          Lichen growth (Werner 1990)
 vars:             8                          14 Jul 2005 14:57
 size:           572 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
locale          str31  %31s                   Locality and feature
point           str1   %9s                    Control point
date            int    %8.0g                  Date
age             int    %8.0g                  Age in years
rshort          float  %9.0g                  Rhizocarpon short axis mm
rlong           float  %9.0g                  Rhizocarpon long axis mm
pshort          int    %8.0g                  P.minuscula short axis mm
plong           int    %8.0g                  P.minuscula long axis mm
-------------------------------------------------------------------------------
Sorted by:
```

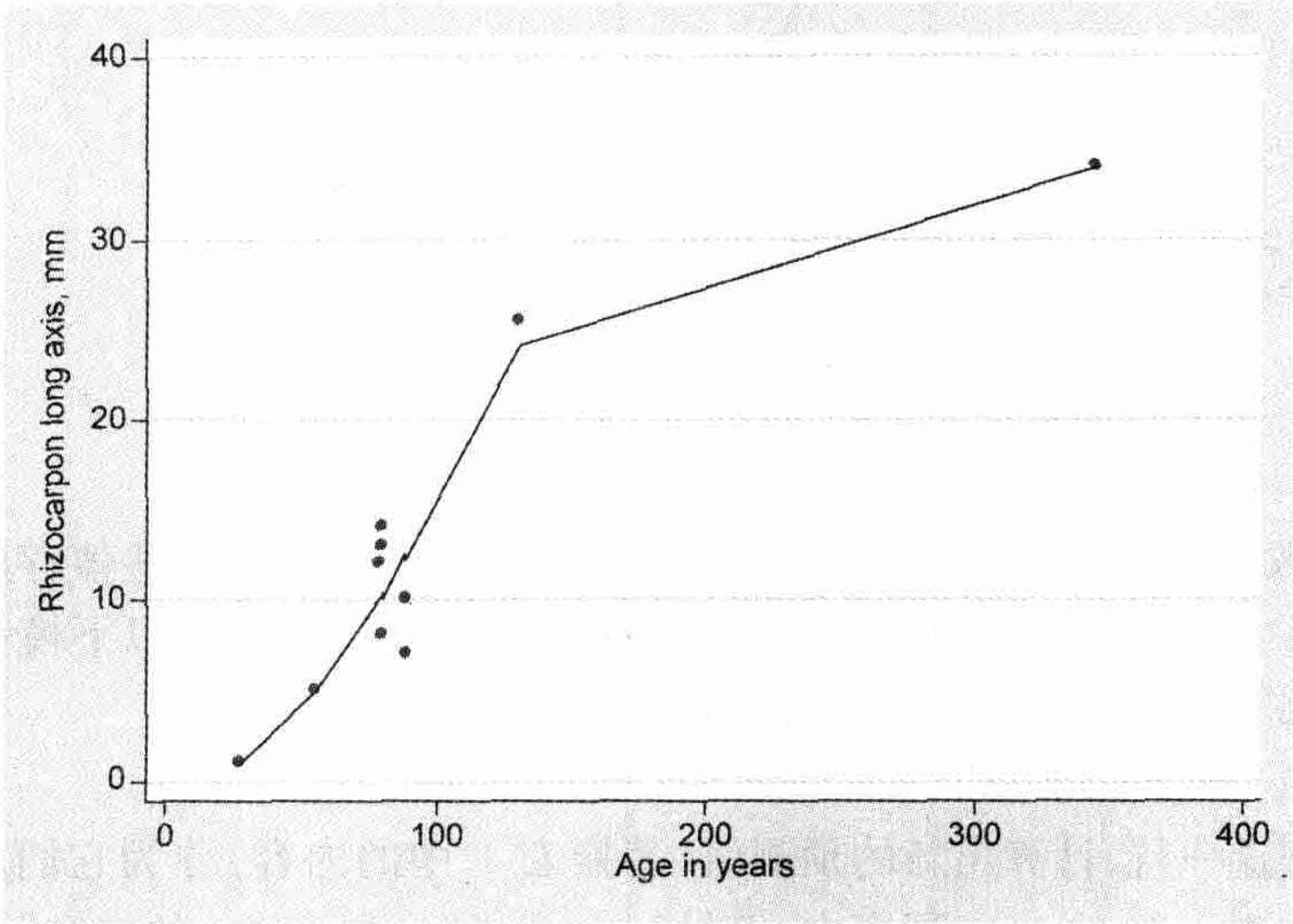


图 8.16

$$y = b_1 exp(-exp(-b_2(x - b_3)))$$

它们可能提供了一个反映苔藓生长过程的合理模型。

如果我们想要画出非线性模型，数据应当包含间隔紧密的 x 取值的合理范围。*lichen.dta* 中11个苔藓样本的实际年龄范围分布从28~346岁。用以下命令，我们可以创建89条另外的人工观测案例，它们的"年龄"以4年为增量从0岁一直到352岁：

```
. range newage 0 396 100
obs was 11, now 100

. replace age = newage[_n-11] if age >= .
(89 real changes made)
```

第一行命令创建一个新变量 *newage*，它包含100个取值，取值范围以4年为增量从0~396。在这样做的过程中，我们也创建了89条新的人工案例，它们在除了 *newage* 之外的所有变量上都是缺失值。**replace** 命令以 *newage* 的取值从0开始替代人工案例中 *age* 的缺失值。数据中的前15条观测案例现在看起来像这样：

```
. list rlong age newage in 1/15
```

```
     +----------------------+
     | rlong   age   newage |
     |----------------------|
  1. |     1    28        0 |
  2. |     5    56        4 |
  3. |    12    79        8 |
  4. |    14    80       12 |
  5. |    13    80       16 |
     |----------------------|
  6. |     8    80       20 |
  7. |     7    89       24 |
  8. |    10    89       28 |
  9. |    34   346       32 |
 10. |    34   346       36 |
     |----------------------|
 11. |  25.5   131       40 |
 12. |     .     0       44 |
 13. |     .     4       48 |
 14. |     .     8       52 |
 15. |     .    12       56 |
     +----------------------+
```

```
. summarize rlong age newage

    Variable |     Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       rlong |      11    14.86364    11.31391          1         34
         age |     100      170.68    104.7042          0        352
      newage |     100         198     116.046          0        396
```

我们现在可以用 **drop** ***newage*** 删除变量 *newage*。只有原始的 11 条观测案例才具有非缺失的 *rlong*(即宿根植物长轴)取值,因此只有它们才将进入模型估计。但是,Stata 对任何一条具有非缺失的 x 取值的观测案例计算预测值。因此,我们既可以得到 11 条真实观测案例的预测值,也可以得到 89 条人工案例的预测值,这些值将为我们准确地画出回归曲线提供了可能。

苔藓生长始于接近于零的尺寸,因此我们选择 **gom3** 这一 Gompertz 函数而不是 **gom4**(它纳入了一个非零的初始水平,即参数 b_0)。图 8.16 表明渐近上限大约为 34,意味着 34 应当是参数 b_1 的合理推测或起始值。这一模型的估计可采用以下命令完成:

```
. nl gom3 rlong age, init(B1=34) nolog
(obs = 11)

      Source |       SS       df       MS            Number of obs =        11
-------------+------------------------------         F(  3,      8) =    125.68
       Model |  3633.16112     3  1211.05371         Prob > F      =    0.0000
    Residual |  77.0888815     8  9.63611018         R-squared     =    0.9792
-------------+------------------------------         Adj R-squared =    0.9714
       Total |     3710.25    11  337.295455         Root MSE      =  3.104208
                                                     Res. dev.     =  52.63435
3-parameter Gompertz function, rlong=b1*exp(-exp(-b2*(age-b3)))
------------------------------------------------------------------------------
       rlong |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          b1 |   34.36637   2.267186    15.16   0.000     29.13823    39.59451
          b2 |   .0217685   .0060806     3.58   0.007     .0077465    .0357904
          b3 |   88.79701   5.632545    15.76   0.000     75.80834    101.7857
------------------------------------------------------------------------------
 (SE's, P values, CI's, and correlations are asymptotic approximations)
```

选项 **nolog** 关闭了结果中对每次迭代的对数似然值的输出显示。所有三个参数的估计值都显著地区别于 1。

我们使用 **predict** 来获得预测值，并画出它们以查看回归曲线。用 **yline** 选项要求显示图 8.17 中这条曲线的下限和估计的上限(0 和 34.366)。

```
. predict yhat
(option yhat assumed; fitted values)

. graph twoway scatter rlong age
      || mspline yhat age, clpattern(solid) bands(50)
      || , legend(off) yline(0 34.366)
      ytitle("Rhizocarpon long axis, mm") xlabel(0(100)400, grid)
```

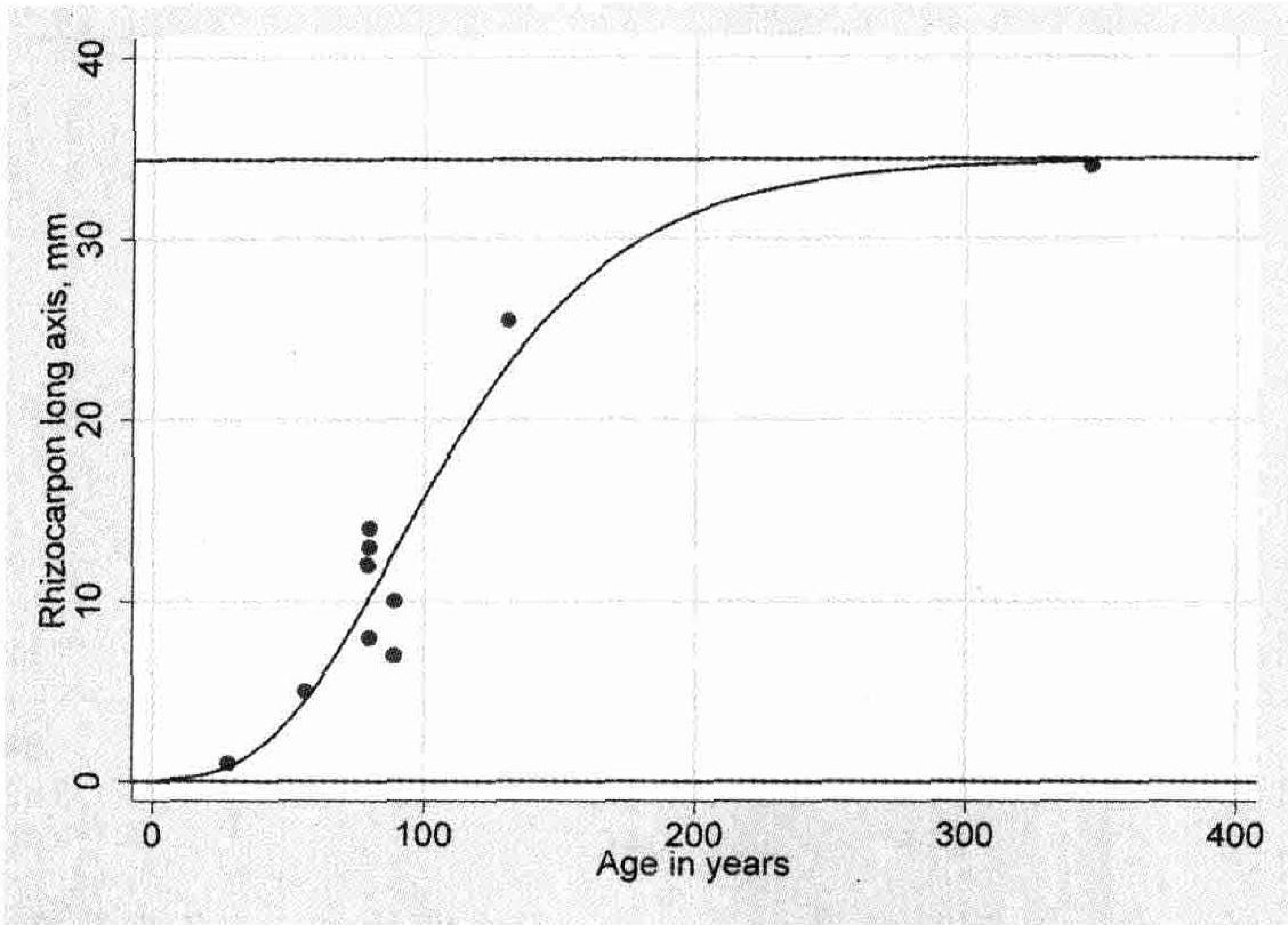


图 8.17

非线性回归程序可能对其初始参数估计值非常敏感，尤其是在处理很少数据或相对复杂的模型时。如果由 nl 函数程序(nl *function*)提供的默认值似乎不起作用的话，**nl** 命令结合 **init** 选项允许研究者建议他们自己的初始值。先前针对类似数据的经验或其他学者发表的成果可能有助于提供合适的初始值。作为替代办法，我们还可以用 **generate** 计算出基于任意选定的数套参数值的预测值以试错方式进行估计，并用 **graph** 标绘图来比较不同预测结果和数据是否一致。

9 稳健回归

Stata 的基本 **regress** 和 **anova** 命令执行常规最小二乘法(OLS)回归。OLS 的普及源自其在假定"理想"数据条件下的理论优势。如果误差分布为正态、独立、同分布(即 normal i.i.d.),那么 OLS 解会比任何其他无偏估计更有效率。然而,这一陈述的反面却常常被忽略了:如果误差不是正态分布,即非 i.i.d.的话,那么其他无偏估计也许比 OLS 做得更好。实际上,OLS 的效率在重尾(heavier-tailed)误差分布(即特异值倾向)条件下迅速退化。然而,这样的分布在许多领域司空见惯。

OLS 倾向于追随特异值,为了拟合它们损失了其他样本案例。长期以来,由于样本中经常包含特异值,进而导致不同样本之间在结果上差异很大,或者说效率较差。稳健回归(robust regression)方法在理想数据条件下几乎可以取得与 OLS 一样的效率,而在数据不理想(比如,误差非正态)时能够取得比 OLS 高得多的效率。"稳健回归"其实包含了多种不同的技术,每一种在处理有问题的数据时都有自己的优点和缺点。本章介绍两种稳健回归,**rreg** 和 **qreg**,并且将它们的结果与 OLS(**regress**)的结果加以比较。

rreg 和 **qreg** 都能抵抗特异值的牵引,在非正态和重尾型误差分布的情况下便能取得高于 OLS 的效率。然而,它们共享 OLS 关于误差独立和同分布的假定。结果是,它们的标准误、统计检验和置信区间在误差异分布或误差相关时也不可信。在使用 **regress** 或其他模型命令时(尽管不是 **rreg** 和 **qreg**),想要放松误差独立和同分布假定,Stata 也提供了估计稳健标准误的选项。

为了简明,本章集中讨论双变量的例子,但是稳健的多元回归或多因素 ANOVA 可以直接应用同样的命令。第 14 章还会回过头来讨论稳健性问题,并说明如何应用蒙特卡罗试验来评价相应的统计技术。

本章描述的几种技术可以从菜单选择上得到:

Statistics-Nonparametric analysis-Quantile regression 分位数回归

Statistics-Linear regression and related-Linear regression-Robust SE 稳健标准误

命令示范

```
. rreg y x1 x2 x3
```

执行 y 对 3 个自变量的稳健回归,采用再加权最小二乘法加上 Huber 和双权数函数,并按 95% 高斯效率调整。在适当设置数据时,**rreg** 还可以取得稳健的平均数、置信区间、平均数差异检验,以及 ANOVA 或 ANCOVA。

```
. rreg y x1 x2 x3, nolog tune(6) genwt(rweight) iterate(10)
```

执行 y 对 3 个自变量的稳健回归。上述选项指示 Stata 不要打印迭代过程的输出，采用调整常数 6(它比默认的 7 能更快使特异值的权数缩小)，产生一个新变量(任意命名为 *rweight*)来为每一案例存放最终迭代的稳健权数，并且限制迭代最多进行 10 次。

. qreg *y x1 x2 x3*

执行 y 对 3 个自变量的分位数回归(quantile regression)，也称为最小绝对值(least absolute value，LAV)回归或最小 *L1* -规范回归(minimum *L1* -norm regression)。按照默认，**qreg** 建立 y 的 0.5 条件分位数(近似于中位数)作为自变量的线性函数，于是提供了一种"中位数回归"。

. qreg *y x1 x2 x3*, quantile(.25)

执行 y 对 3 个自变量的分位数回归，建立 y 的 0.25 条件分位数(第一四分位)作为 *x1*、*x2*、*x3* 的线性函数。

. bsqreg *y x1 x2 x3*, rep(100)

执行分位数回归，用自助法(bootstrap)对数据重复抽样 100 遍(默认设置为 **rep(20)**)以估计出标准误。

. predict *e*, resid

在执行 **regress**、**rreg**、**qreg**、**bsqreg** 命令之后，进一步计算出残差值(指定命名为 e)。与此类似，**predict yhat** 可进一步计算出 y 的预测值。其他 **predict** 选项在某些限制条件下也可应用。

. regress *y x1 x2 x3*, robust

执行 y 对 3 个自变量的 OLS 回归。通过不需要假定误差同分布的稳健方法(Huber/White方法或三明治方法)对系数的方差及标准误进行估计。如果加上 **cluster( )** 选项，还可容纳误差之间一种来源的相关。《用户指南》中描述了这些方法背后的原理。

用理想数据的回归

为了阐明稳健性问题，我们来探究一个人工小数据($n=20$)*robust1.dta*：

```
Contains data from C:\data\robust1.dta
  obs:            20                          Robust regression examples 1
                                                (artificial data)
 vars:            10                          17 Jul 2005 09:35
 size:           880 (99.9% of memory free)

-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
x               float  %9.0g                  Normal X
e1              float  %9.0g                  Normal errors
y1              float  %9.0g                  y1 = 10 + 2*x + e1
e2              float  %9.0g                  Normal errors with 1 outlier
y2              float  %9.0g                  y2 = 10 + 2*x + e2
x3              float  %9.0g                  Normal X with 1 leverage obs.
e3              float  %9.0g                  Normal errors with 1 extreme
y3              float  %9.0g                  y3 = 10 + 2*x3 + e3
e4              float  %9.0g                  Skewed errors
y4              float  %9.0g                  y4 = 10 + 2*x + e4
-------------------------------------------------------------------------------
Sorted by:
```

变量 x 和 $e1$ 各自都包括 20 个来自独立标准正态分布的随机值。$y1$ 包括 20 个由回归模型产生的值：

$$y1 = 10 + 2x + e1$$

形成这前 3 个变量的命令为：

```
. clear
. set obs 20
. generate x = invnorm(uniform())
. generate e1 = invnorm(uniform())
. generate y1 = 10 + 2*x + e1
```

要是用实际数据，编码错误和测量误差有时会导致极特异的值。为了模拟这种情况，我们可以将第 2 个案例的误差从 -0.89 改为 19.89：

```
. generate e2 = e1
. replace e2 = 19.89 in 2
. generate y2 = 10 + 2*x + e2
```

用类似的处理方法形成了 *robust1 .dta* 中的其他一些变量。

y1 和 *x* 呈现了一种理想的回归问题：*y1* 的期望值其实只是 *x* 的线性函数，误差来自于正态、独立，并且相同的分布，因为这些都是我们定义的。OLS 很好地估计了真实的截距(10)和斜率(2)，取得的回归线如图 9.1 所示。

```
. regress y1 x

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =  108.25
       Model |  134.059351     1  134.059351           Prob > F      =  0.0000
    Residual |    22.29157    18  1.23842055           R-squared     =  0.8574
-------------+------------------------------           Adj R-squared =  0.8495
       Total |  156.350921    19  8.22899586           Root MSE      =  1.1128

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   2.048057   .1968465    10.40   0.000     1.634498    2.461616
       _cons |   9.963161   .2499861    39.85   0.000      9.43796    10.48836
------------------------------------------------------------------------------
```

用迭代再加权最小二乘法(IRLS)程序 **rreg** 来取得稳健回归估计。第一步 **rreg** 迭代是从 OLS 估计开始的。在第一步后任何影响大到 Cook 的 *D* 值大于 1 的案例都将会自动被搁置一边。然后，应用 Huber 函数为每一个案例计算出权数，它会使残差较大的案例得到较小的权数，再继续进行加权最小二乘法(WLS)估计。经过几步 WLS 迭代，权数函数转变为 Tukey 双权(Tukey biweight，参见 Li，1985)，并按 95% 高斯效率加以调整(细节参见 Hamilton，1992a)。**rreg** 估计出标准误并进行假设检验，用的是伪值法(pseudo values method)(见 Street，Carroll 和 Ruppert，1988)，因为其不需要假定正态性。

这一"理想数据"示例没有包括严重的特异值，所以这里本来用不着 **rreg**。**rreg** 所取得的截距和斜率估计与 **regress** 所取得的类似(都与真值 10 和 2 差得不多)，但是它们的估计标准误稍大一点。在 normal i.i.d.误差条件下，正如本例所示，**rreg** 理论上拥有 OLS 效率的 95% 。

rreg 与 **regress** 都同属最大似然估计族(*M*-estimators)。而另一种序次统计估计法(*L*-estimators)采用拟合 *y* 的分位数，而不是它的期望值或平均数。比如，我们可以建模表示 *y* 的中位数(0.5 分位数)如何随 *x* 变化。**qreg**，一种 *L1* 型估计，可以

完成这种分位数回归,并提供了另一种能够很好抵抗特异值的方法:

```
. predict yhatlo

. graph twoway scatter y1 x
      || line yhatlo x, clpattern(solid) sort
      || , ytitle("y1 = 10 + 2*x + e1") legend(order(2)
         label(2 "OLS line") position(11) ring(0) cols(1))
```

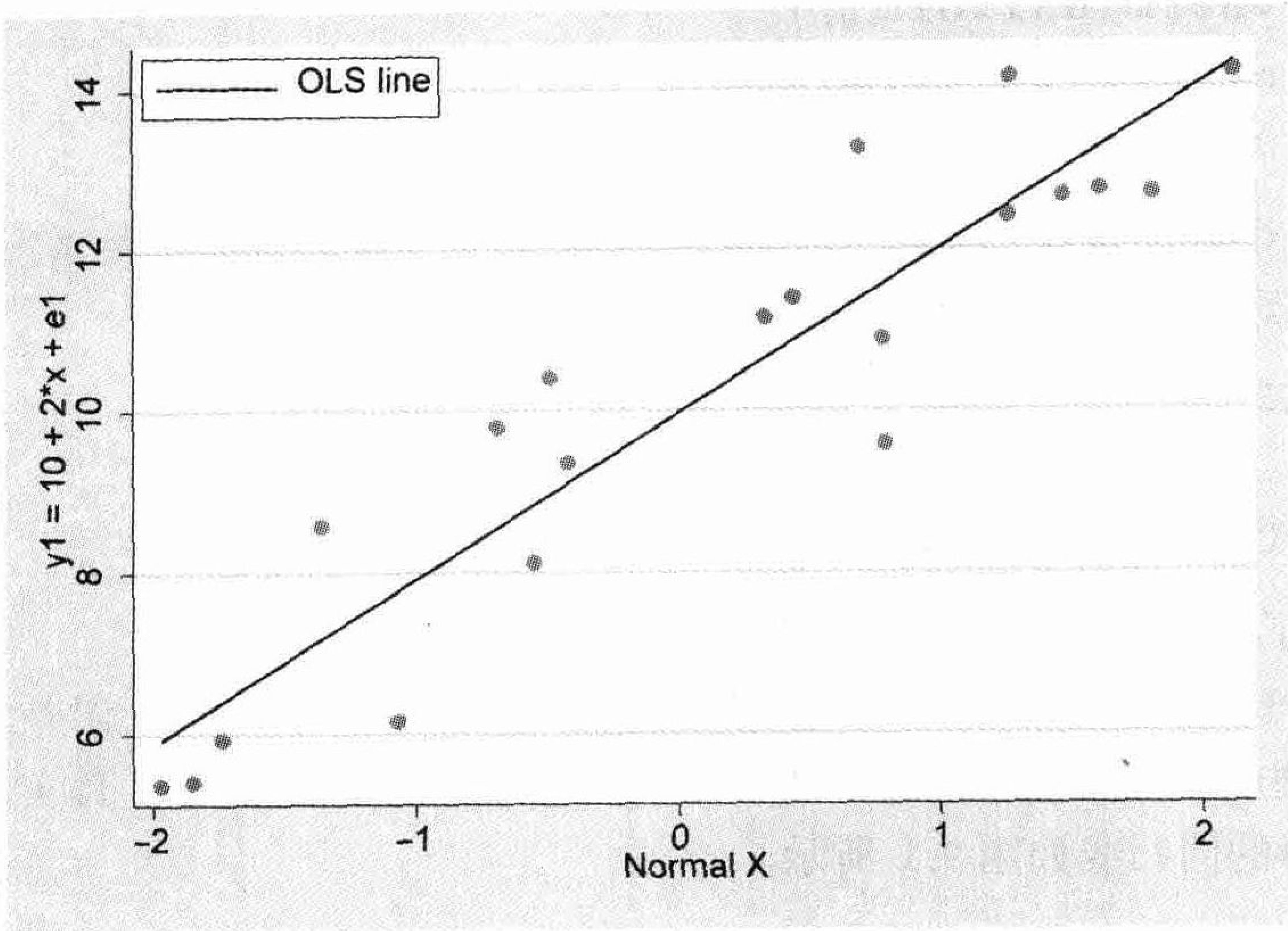


图 9.1

```
. rreg y1 x

   Huber iteration 1:  maximum difference in weights = .35774407
   Huber iteration 2:  maximum difference in weights = .02181578
Biweight iteration 3:  maximum difference in weights = .14421371
Biweight iteration 4:  maximum difference in weights = .01320276
Biweight iteration 5:  maximum difference in weights = .00265408

Robust regression estimates                          Number of obs =      20
                                                     F(  1,    18) =   79.96
                                                     Prob > F      =  0.0000

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   2.047813   .2290049     8.94   0.000     1.566692    2.528935
       _cons |   9.936163   .2908259    34.17   0.000     9.325161    10.54717
------------------------------------------------------------------------------

. qreg y1 x

Iteration  1:  WLS sum of weighted deviations =  17.711531

Iteration  1: sum of abs. weighted deviations =  17.130001
Iteration  2: sum of abs. weighted deviations =  16.858602

Median regression                                   Number of obs =       20
  Raw sum of deviations     46.84 (about 10.4)
  Min sum of deviations   16.8586                   Pseudo R2     =   0.6401

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   2.139896   .2590447     8.26   0.000     1.595664    2.684129
       _cons |    9.65342   .3564108    27.09   0.000     8.904628    10.40221
------------------------------------------------------------------------------
```

尽管 **qreg** 取得了合理的参数估计,但它们的标准误都超过了 **regress**(OLS)和 **rreg**。在理想数据条件下,**qreg** 是这 3 种估计中效率最差的。在以下各节,我们再来看看它们

在数据并不理想条件下的作为。

Y 上的特异值

变量 *y2* 与 *y1* 相同,但是有一个由第 2 号案例的“严重”误差所导致的特异值。OLS 估计对特异值几乎没有什么抵抗力,所以案例 2 的这一变化(在图 9.2 的左上部)极大地改变了 **regress** 的结果:

```
. regress y2 x

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =    0.97
       Model |   18.764271     1   18.764271           Prob > F      =  0.3378
    Residual |  348.233471    18  19.3463039           R-squared     =  0.0511
-------------+------------------------------           Adj R-squared = -0.0016
       Total |  366.997742    19  19.3156706           Root MSE      =  4.3984

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   .7662304   .7780232     0.98   0.338    -.8683356    2.400796
       _cons |    11.1579   .9880542    11.29   0.000     9.082078    13.23373
------------------------------------------------------------------------------

. predict yhat2o
(option xb assumed; fitted values)

. label variable yhat2o "OLS line (regress)"
```

这个特异值提高了 OLS 截距(从 9.936 升至 11.157 9),并且减低了斜率(从 2.048 降到 0.766)。R^2 也从 0.857 4 减小到 0.051 1。标准误相当于原来的 4 倍,而且 OLS 斜率(图 9.2 中的实线)变得不再显著区别于 0 了。

然而,正如图 9.2 中虚线所示,这个特异值对于 **rreg** 几乎没什么影响。稳健系数几乎没什么变化,仍然接近于真实参数 10 和 2,并且稳健标准误也没有提高多少。

```
. rreg y2 x, nolog genwt(rweight2)

Robust regression estimates                            Number of obs =      19
                                                       F(  1,    17) =   63.01
                                                       Prob > F      =  0.0000

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.979015   .2493146     7.94   0.000     1.453007    2.505023
       _cons |   10.00897   .3071265    32.59   0.000     9.360986    10.65695
------------------------------------------------------------------------------

. predict yhat2r
(option xb assumed; fitted values)

. label variable yhat2r "robust regression (rreg)"

. graph twoway scatter y2 x
      ||  line yhat2o x, clpattern(solid) sort
      ||  line yhat2r x, clpattern(longdash) sort
      ||  , ytitle("y2 = 10 + 2*x + e2")
      legend(order(2 3) position(1) ring(0) cols(1) margin(sides))
```

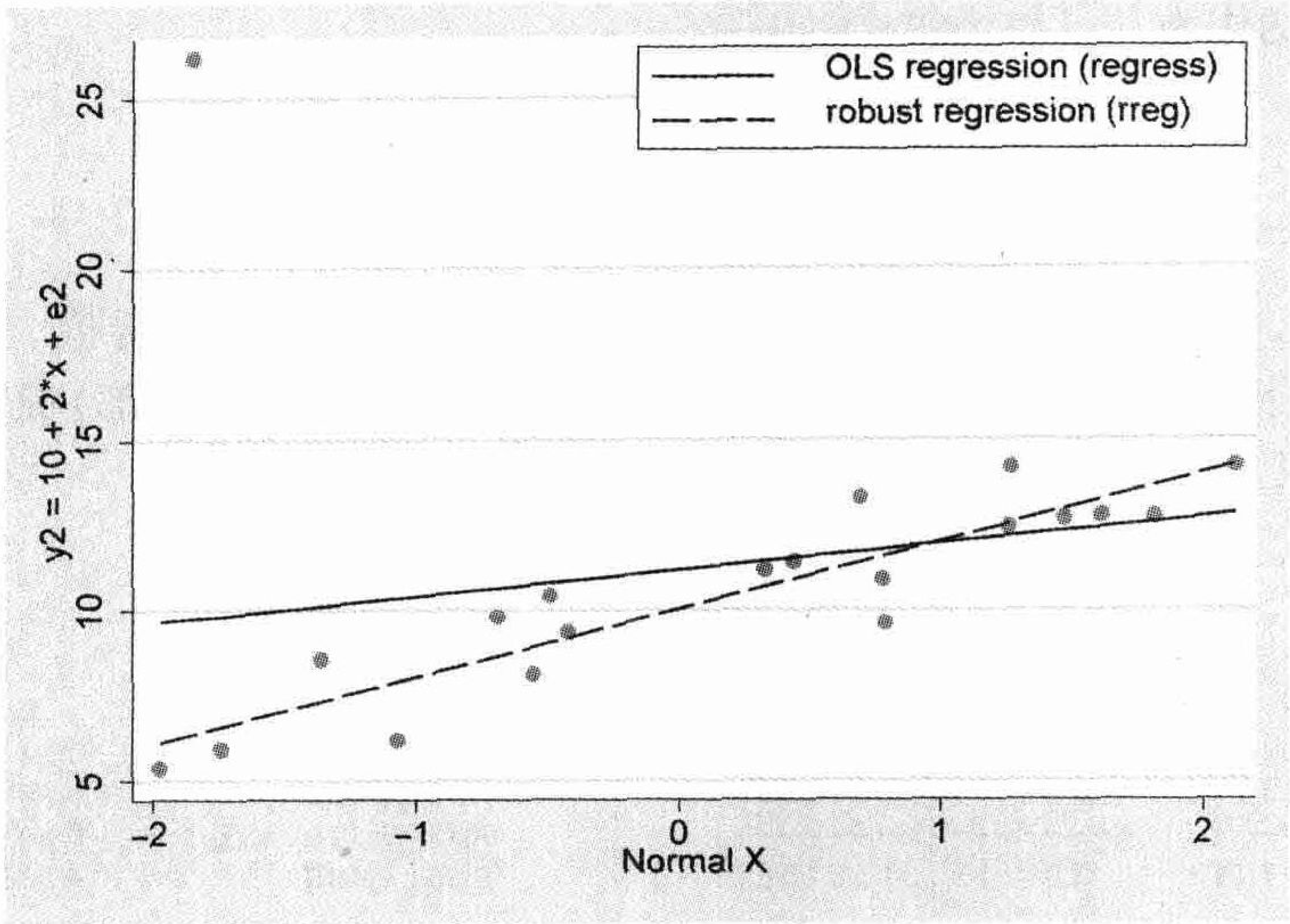


图 9.2

上述命令的 **nolog** 选项导致 Stata 不再打印迭代记录。选项 **genwt（*rweight2*）** 将稳健权数存为名为 *rweight2* 的变量。

```
. predict resid2r, resid
. list y2 x resid2r rweight2
```

	y2	x	resid2r	rweight2
1.	5.37	-1.97	-.7403071	.94644465
2.	26.19	-1.85	19.84221	.
3.	5.93	-1.74	-.6354806	.96037073
4.	8.58	-1.36	1.262494	.8493384
5.	6.16	-1.07	-1.731421	.7257631
6.	9.80	-0.69	1.156554	.87273631
7.	8.12	-0.55	-.8005085	.93758391
8.	10.40	-0.49	1.36075	.82606386
9.	9.35	-0.42	.17222	.99712388
10.	11.16	0.33	.4979582	.97581674
11.	11.40	0.44	.5202664	.97360863
12.	13.26	0.69	1.885513	.68048066
13.	10.88	0.78	-.6725982	.95572833
14.	9.58	0.79	-1.992389	.64644918
15.	12.41	1.26	-.0925257	.99913568
16.	14.14	1.27	1.617685	.75887073
17.	12.66	1.47	-.2581189	.99338589
18.	12.74	1.61	-.4551811	.97957817
19.	12.70	1.81	-.8909839	.92307041
20.	14.19	2.12	-.0144787	.99997651

接近于 0 的残差所产生的权数接近于 1，越大的残差得到越发更小的权数。案例 2 由于影响过大已经被自动地搁置一边，因为其 Cook 的 *D* 统计量已经大于 1 了，所以 **rreg** 分配给案例 2 的权数为“缺失”，于是这个案例对最终估计完全没有影响。要是用 **regress** 伴以分析权数的回归（结果略）会得到相同的最终估计，但是标准误或统计检验是不正确的：

```
. regress y2 x [aweight = rweight2]
```

要是用 **qreg** 做 *y2* 对 *x* 的回归,也能抵抗特异值的影响,并且比 **regress** 做得要更好,但是其表现不如 **rreg** 。**qreg** 显得比 **rreg** 的效率低,并且就这个样本的系数估计距离真值 10 和 2 来说有点过大。

```
. qreg y2 x, nolog

Median regression                                    Number of obs =        20
  Raw sum of deviations     56.68 (about 10.88)
  Min sum of deviations 36.20036                     Pseudo R2     =    0.3613

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.821428   .4105944     4.44   0.000     .9588014    2.684055
       _cons |     10.115   .5088526    19.88   0.000     9.045941    11.18406
------------------------------------------------------------------------------
```

蒙特卡罗研究者也已经注意到,用 **qreg** 计算的标准误有时会低估计真正的样本之间的变异,尤其是当样本规模较小时。作为一种替换,Stata 提供了 **bsqreg** 命令,它与 **qreg** 完成同样的中位数或分位数回归,但运用自助法(bootstrapping,即数据再抽样)来估计标准误。选项 **rep( )** 控制重复的次数。它的默认设置是 **rep(20)**,这对于探测性工作已经足够了。在取得"最终"结论之前,我们可以多花点时间抽出 200 或更多的自助样本。**qreg** 和 **bsqreg** 拟合的是同样的模型。在下面的例子中, **bsqreg** 也取得了类似的标准误。到第 14 章时我们还会再谈自助法的话题。

```
. bsqreg y2 x, rep(50)

(fitting base model)
(bootstrapping ..................................................)

Median regression, bootstrap(50) SEs                 Number of obs =        20
  Raw sum of deviations     56.68 (about 10.88)
  Min sum of deviations 36.20036                     Pseudo R2     =    0.3613

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.821428   .4084728     4.46   0.000     .9632587    2.679598
       _cons |     10.115   .4774718    21.18   0.000     9.111869    11.11813
------------------------------------------------------------------------------
```

X 上的特异值(杠杆作用)

rreg 、**qreg** 、**bsqreg** 都能较好地处理 *y* 上的特异值,除非具有异常 *y* 值的案例还同时具有异常的 *x* 值(也称杠杆作用,leverage)。在 *robust1.dta* 数据中的变量 *y3* 和 *x3* 提供了关于杠杆的极端例子。除案例 2 是个杠杆作用案例以外,其他所有变量值都与 *y1* 和 *x* 相同。

案例 2 有很强的杠杆作用,再加上它有非寻常的 *y3* 值,两者结合起来导致其影响巨大:**regress** 和 **qreg** 都追随这个特异值,报告说"最佳拟合"线有负的斜率(图 9.3)。

```
. regress y3 x3

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =   11.01
       Model |  139.306724     1  139.306724           Prob > F      =  0.0038
    Residual |  227.691018    18   12.649501           R-squared     =  0.3796
-------------+------------------------------           Adj R-squared =  0.3451
       Total |  366.997742    19  19.3156706           Root MSE      =  3.5566

------------------------------------------------------------------------------
          y3 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x3 |  -.6212248   .1871973    -3.32   0.004    -1.014512   -.227938
       _cons |   10.80931   .8063436    13.41   0.000     9.115244   12.50337
------------------------------------------------------------------------------

. predict yhat3o

. label variable yhat3o "OLS regression (regress)"

. qreg y3 x3, nolog

Median regression                                    Number of obs =        20
  Raw sum of deviations     56.68 (about 10.88)
  Min sum of deviations 56.19466                     Pseudo R2     =    0.0086

------------------------------------------------------------------------------
          y3 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x3 |  -.6222217    .347103    -1.79   0.090    -1.351458   .1070146
       _cons |   11.36533   1.419214     8.01   0.000     8.383676   14.34699
------------------------------------------------------------------------------

. predict yhat3q

. label variable yhat3q "median regression (qreg)"

. rreg y3 x3, nolog
Robust regression estimates                            Number of obs =      19
                                                       F(  1,    17) =   63.01
                                                       Prob > F      =  0.0000

------------------------------------------------------------------------------
          y3 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x3 |   1.979015   .2493146     7.94   0.000     1.453007    2.505023
       _cons |   10.00897   .3071265    32.59   0.000     9.360986    10.65695
------------------------------------------------------------------------------

. predict yhat3r

. label variable yhat3r "robust regression (rreg)"

. graph twoway scatter y3 x3
        ||  line yhat3o x3, clpattern(solid) sort
        ||  line yhat3r x3, clpattern(longdash) sort
        ||  line yhat3q x3, clpattern(shortdash) sort ,
       ytitle("y3 = 10 + 2*x + e3") legend(order(4 3 2) position(5)
          ring(0) cols(1) margin(sides)) ylabel(-30(10)30)
```

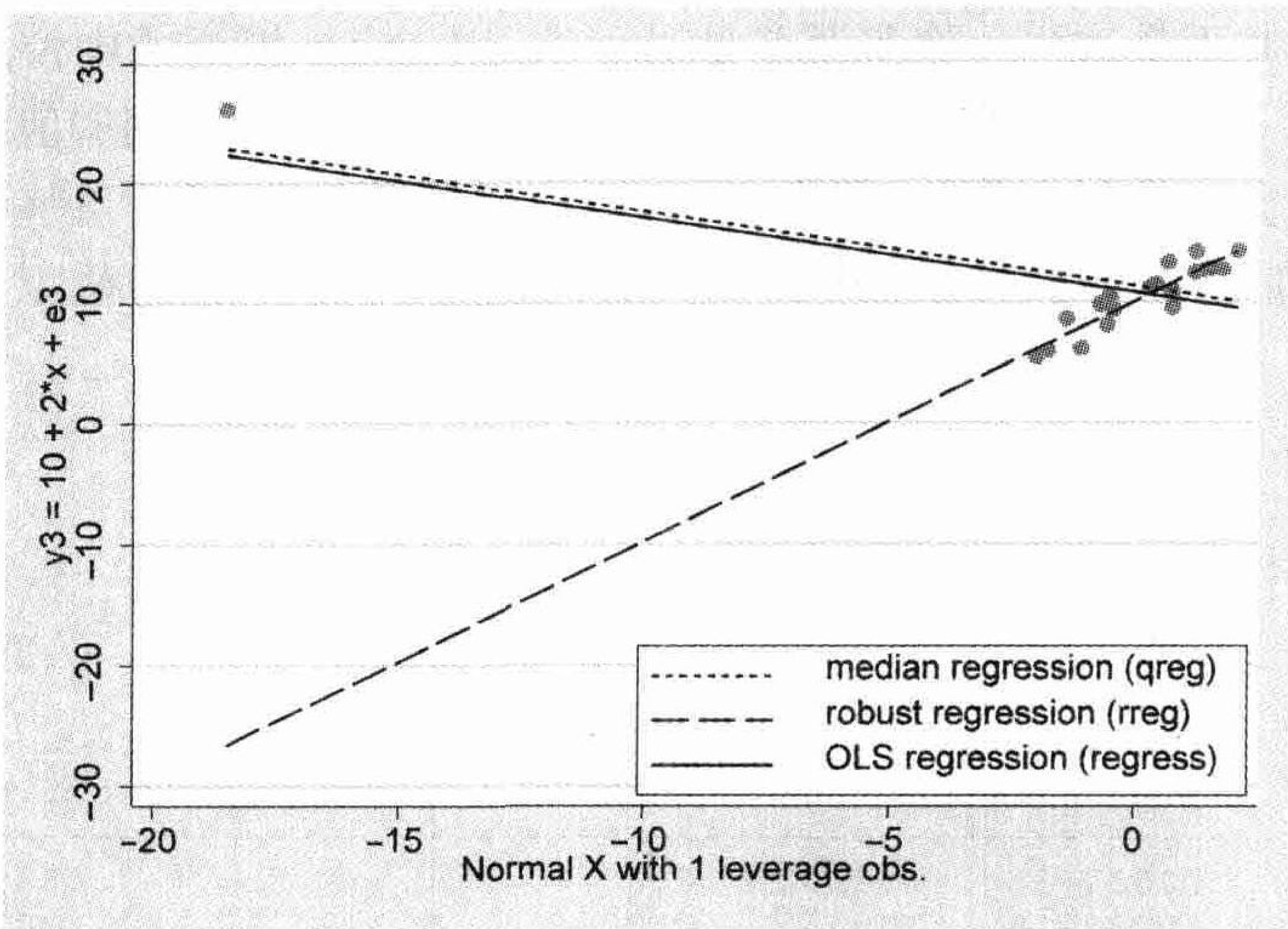


图 9.3

图 9.3 显示出,**regress** 和 **qreg** 对于杠杆作用(即 *x* 上特异值)并不稳健。然而,**rreg** 程序不仅削弱了较大残差案例的权重(这种功能本身并不能防护杠杆影响),而且还自动地将那些 Cook 的 *D*(影响)统计量大于 1 的案例搁置在外了。当我们将 *y3* 对 *x3* 回归时,这种情况就发生了。**rreg** 不再理睬这个最有影响的观测案例,在其他 19 个案例基础上求出了一条更加合理的正斜率的回归线。

将影响特大的案例置于不顾,就像 **rreg** 所为,提供了一种简单的但是并不十分安全的方式来处理杠杆作用。还存在着更综合的方法,称为有限影响回归(bounded-influence regression),也可以在 Stata 程序中执行。

图 9.2 和图 9.3 的例子只涉及了单一特异值,其实稳健程序可以处理更多特异值。如果有太多严重的特异值,或者有一组类似的特异值,可能会导致稳健程序中止。但是在这种场合,诊断用的标绘图常常值得加以注意,分析人员必须要问,拟合一个线性模型是否有意义。很可能值得去寻求一种明确的模型来解释什么导致这些特异值之所以特异。

蒙特卡罗试验(在第 14 章示范)确认,像 **rreg** 和 **qreg** 这样的估计方法应用于重尾(特异值倾向)但对称的误差分布时,通常能保持无偏,效率要优于 OLS 估计。下一节示范当误差为不对称分布时会产生什么结果。

不对称的误差分布

在数据 *robust1.dta* 中,变量 *e4* 呈偏态分布并含有特异值:*e4* 等于将 *e1*(标准正态变量)做 4 次方、然后调整为平均值为 0。这些偏态误差、加上与 *x* 之间的线性关系定义变量 *y4* = 10 + 2*x* + *e4*。不管误差分布的形状如何,OLS 仍然是无偏估计。从趋向上看,其估计应该以真实参数值为中心。

```
. regress y4 x
      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =    6.97
       Model |  155.870383     1  155.870383           Prob > F      =  0.0166
    Residual |  402.341909    18  22.3523283           R-squared     =  0.2792
-------------+------------------------------           Adj R-squared =  0.2392
       Total |  558.212291    19  29.3795943           Root MSE      =  4.7278

------------------------------------------------------------------------------
          y4 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   2.208388   .8362862     2.64   0.017     .4514157     3.96536
       _cons |   9.975681   1.062046     9.39   0.000     7.744406    12.20696
------------------------------------------------------------------------------
```

但是大多数稳健估计却并不是这样。除非误差是对称的,用 **qreg** 拟合的中位线或用 **rreg** 拟合的双权(biweight)线在理论上并不与用 **regress** 估计的 *y* 期望值线相符。只要偏态误差只反映在分布中很小部分,那么 **rreg** 展示不出有偏。但是当整个分布都呈偏态时,比如,像 *e4* 那样,**rreg** 就会集中在一侧削弱权数,导致 *y* 上的截距估计显著有偏。

```
. rreg y4 x, nolog

Robust regression estimates                        Number of obs =        20
                                                   F(  1,    18) =   1319.29
                                                   Prob > F      =    0.0000

------------------------------------------------------------------------------
          y4 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.952073   .0537435    36.32   0.000     1.839163    2.064984
       _cons |   7.476669   .0682518   109.55   0.000     7.333278    7.620061
------------------------------------------------------------------------------
```

在图 9.4 中,尽管 **rreg** 取得的 *y* 截距过低,其斜率却与 OLS 线和真实模型保持平行。其实,由于受特异值影响较少,**rreg** 的斜率(1.95)更接近于真实斜率(2),并且其标准误也比 **regress** 结果小得多。这就表明,在使用 **rreg** 或类似估计方法于偏态误差数据时要有所权衡:在 *y* 截距估计上存在有偏风险,但是回归系数估计可望无偏、并相对更精确。在许多研究场合,斜率比截距更有意义,因此这种得失是值得的。此外,稳健的 *t* 检验和 *F* 检验中并不需要假定正态误差,这与 OLS 估计中有所不同。

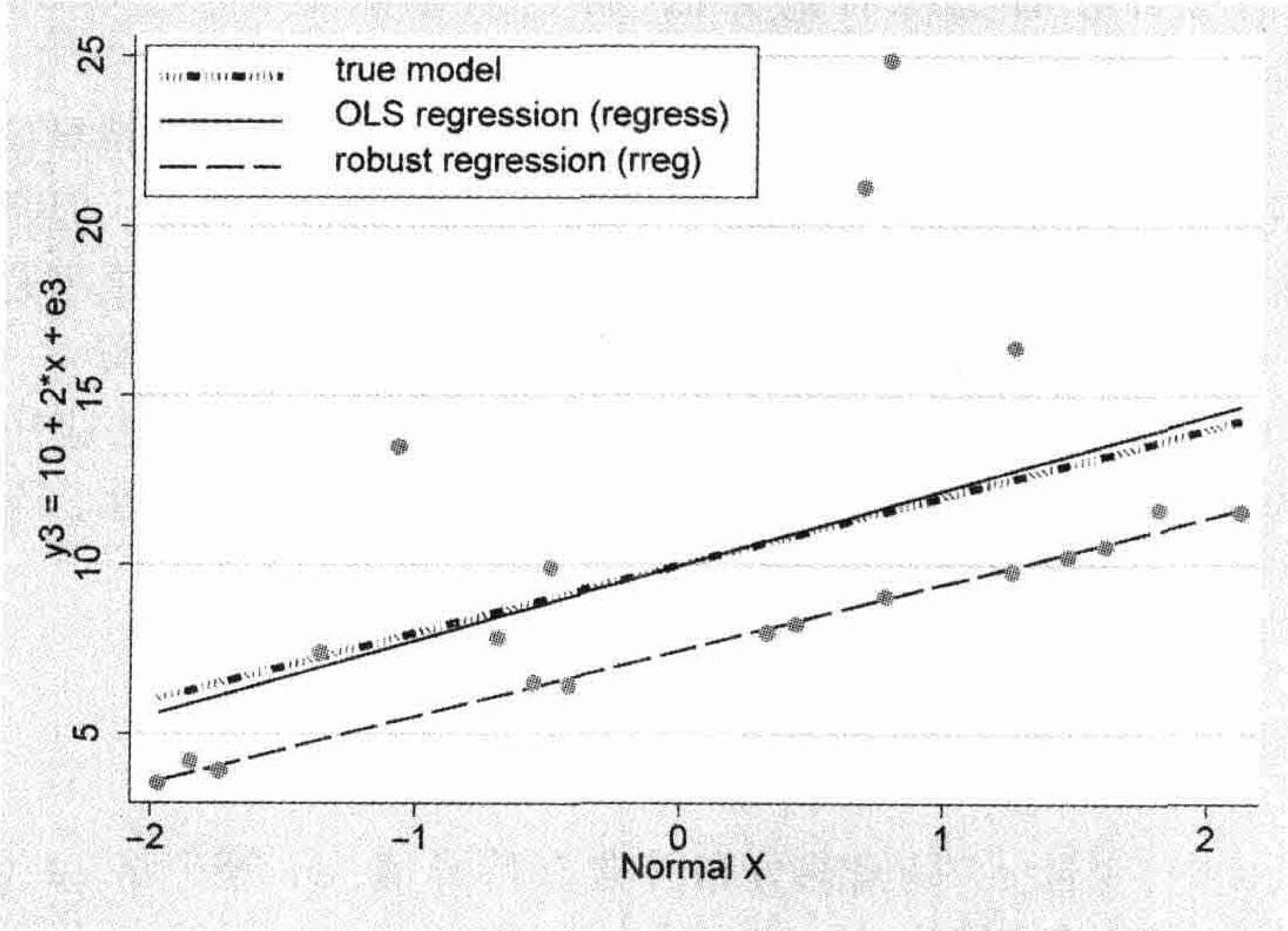


图 9.4

稳健的方差分析

一旦方差分析模型改用回归形式,**rreg** 还能用于稳健的方差分析或协方差分析。我们用某校教师工资数据 *faculty.dta* 来加以示范。

教师工资是随职称而提高的。在这个数据中,男性有更高的平均工资水平:

常规(OLS)的方差分析表明,职称 *rank* 与性别 *gender* 都对工资有显著影响。

但是工资并不是正态分布,并且高级职称平均工资反映出可能有特异值影响,即有少数人工资极高。假如我们想要通过稳健的方差分析来检查这些结果。我们需要 *rank* 与 *gender* 的相应效应编码(effect-coding)变量,这一数据也已经包括了:

```
Contains data from C:\data\faculty.dta
  obs:           226                          College faculty salaries
 vars:             6                          17 Jul 2005 09:32
 size:         2 938 (99.9% of memory free)

-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
rank            byte   %8.0g       rank       Academic rank
gender          byte   %8.0g       sex        Gender (dummy variable)
female          byte   %8.0g                  Gender (effect coded)
assoc           byte   %8.0g                  Assoc Professor (effect coded)
full            byte   %8.0g                  Full Professor (effect coded)
pay             float  %9.0g                  Annual salary
-------------------------------------------------------------------------------
Sorted by:
```

. table *gender rank*, contents(mean *pay*)

```
----------------------------------------
Gender     |
(dummy     |         Academic rank
variable)  |   Assist     Assoc      Full
-----------+----------------------------
      Male |    29280  38622.22   52084.9
    Female | 28711.04  38019.05     47190
----------------------------------------
```

. anova *pay rank gender rank*gender*

```
                   Number of obs =     226     R-squared     =  0.7305
                   Root MSE      = 5108.21     Adj R-squared =  0.7244

       Source |  Partial SS    df       MS           F     Prob > F
  ------------+----------------------------------------------------
        Model |  1.5560e+10     5  3.1120e+09     119.26     0.0000
              |
         rank |  7.6124e+09     2  3.8062e+09     145.87     0.0000
       gender |   127361829     1   127361829       4.88     0.0282
  rank*gender |  87997720.1     2  43998860.1       1.69     0.1876
              |
     Residual |  5.7406e+09   220  26093824.5
  ------------+----------------------------------------------------
        Total |  2.1300e+10   225  94668810.3
```

. tabulate *gender female*

```
    Gender |
    (dummy | Gender (effect coded)
 variable) |        -1          1 |     Total
-----------+----------------------+----------
      Male |       149          0 |       149
    Female |         0         77 |        77
-----------+----------------------+----------
     Total |       149         77 |       226
```

```
. tabulate rank assoc

  Academic |  Assoc Professor (effect coded)
      rank |        -1          0          1 |     Total
-----------+---------------------------------+----------
    Assist |        64          0          0 |        64
     Assoc |         0          0        105 |       105
      Full |         0         57          0 |        57
-----------+---------------------------------+----------
     Total |        64         57        105 |       226
```

```
. tab rank full

  Academic |  Full Professor (effect coded)
      rank |        -1          0          1 |     Total
-----------+---------------------------------+----------
    Assist |        64          0          0 |        64
     Assoc |         0        105          0 |       105
      Full |         0          0         57 |        57
-----------+---------------------------------+----------
     Total |        64        105         57 |       226
```

如果 *faculty.dta* 中并没有这些效应编码变量(如 *female*、*assoc* 和 *full*),我们可以根据 *gender* 和 *rank* 的信息用一系列 **generate** 和 **replace** 命令来建立。另外,我们还需要建立两个交互项(interaction terms)来代表女性副教授和女性正教授:

```
. generate femassoc = female*assoc
. generate femfull = female*full
```

男性和助理教授在这个例子中都属于"省略类型"。现在我们就可以用回归来完成以前所作的方差分析了:

```
. regress pay assoc full female femassoc femfull

      Source |       SS       df       MS              Number of obs =     226
-------------+------------------------------           F(  5,   220) =  119.26
       Model |  1.5560e+10     5  3.1120e+09           Prob > F      =  0.0000
    Residual |  5.7406e+09   220  26093824.5           R-squared     =  0.7305
-------------+------------------------------           Adj R-squared =  0.7244
       Total |  2.1300e+10   225  94668810.3           Root MSE      =  5108.2

------------------------------------------------------------------------------
         pay |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       assoc |  -663.8995   543.8499    -1.22   0.223    -1735.722    407.9229
        full |   10652.92   783.9227    13.59   0.000     9107.957    12197.88
      female |  -1011.174   457.6938    -2.21   0.028    -1913.199   -109.1483
    femassoc |   709.5864   543.8499     1.30   0.193    -362.2359    1781.409
     femfull |  -1436.277   783.9227    -1.83   0.068    -2981.236    108.6819
       _cons |   38984.53   457.6938    85.18   0.000     38082.51    39886.56
------------------------------------------------------------------------------
```

```
. test assoc full

 ( 1)  assoc = 0.0
 ( 2)  full = 0.0

       F(  2,   220) =  145.87
            Prob > F =    0.0000
```

```
. test female

 ( 1)   female = 0.0

       F(  1,   220) =    4.88
            Prob > F =    0.0282

. test femassoc femfull

 ( 1)   femassoc = 0.0
 ( 2)   femfull = 0.0

       F(  2,   220) =    1.69
            Prob > F =    0.1876
```

执行 **regress** 之后再执行适当的 **test** 命令就能取得与我们以前 **anova** 同样的 R^2 和 F 检验结果，这里回归预测值就等于平均工资。

```
. predict predpay1
(option xb assumed; fitted values)

. label variable predpay1 "OLS predicted salary"

. table gender rank, contents(mean predpay1)

----------------------------------------
Gender     |
(dummy     |          Academic rank
variable)  |   Assist     Assoc       Full
-----------+----------------------------
      Male |    29280  38622.22    52084.9
    Female | 28711.04  38019.05      47190
----------------------------------------
```

预测值(即平均数)、R^2 和 F 检验的结果并不取决于我们在回归中省略了哪个类别，因为所谓的“省略类别”，男性与助理教授，在回归并没有真的省略。它们的信息暗含于所包括的类别中：即如果一个教师不是女的，那么他就一定是男的，以此类推。

为了完成稳健的方差分析，就应用 **rreg** 于这个模型：

```
. rreg pay assoc full female femassoc femfull, nolog

Robust regression estimates                        Number of obs =      226
                                                   F(  5,   220) =   138.25
                                                   Prob > F      =   0.0000

------------------------------------------------------------------------------
         pay |      Coef.   Std. Err.       t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       assoc |  -315.6463   458.1588     -0.69   0.492    -1218.588    587.2956
        full |   9765.296   660.4048     14.79   0.000     8463.767    11066.83
      female |  -749.4949   385.5778     -1.94   0.053    -1509.394    10.40395
    femassoc |   197.7833   458.1588      0.43   0.666    -705.1587    1100.725
     femfull |   -913.348   660.4048     -1.38   0.168    -2214.878    388.1815
       _cons |   38331.87   385.5778     99.41   0.000     37571.97    39091.77
------------------------------------------------------------------------------

. test assoc full

 ( 1)   assoc = 0.0
 ( 2)   full = 0.0

       F(  2,   220) =  182.67
            Prob > F =    0.0000
```

```
. test female

 ( 1)  female = 0.0

       F(  1,   220) =    3.78
            Prob > F =    0.0532

. test femassoc femfull

 ( 1)  femassoc = 0.0
 ( 2)  femfull = 0.0

       F(  2,   220) =    1.16
            Prob > F =    0.3144
```

rreg 削弱了几个特异值的权数，主要是那些高薪的男性正教授。要看稳健平均数，就再次使用预测值：

```
. predict predpay2
(option xb assumed; fitted values)

. label variable predpay2 "Robust predicted salary"

. table gender rank, contents(mean predpay2)

----------------------------------------
Gender     |
(dummy     |        Academic rank
variable)  |   Assist      Assoc       Full
----------+-----------------------------
      Male | 28916.15   38567.93   49760.01
    Female | 28848.29   37464.51   46434.32
----------------------------------------
```

要是我们看稳健平均数，那么在助理教授和正教授内的男女差异显得较小，尽管并没有完全消失。但是，副教授内的性别差异却有少许扩大。

辅以效应编码和适当的交互项，**regress** 可以准确重现方差分析结果。**rreg** 也能完成类似的分析，但检验的是稳健平均数（而不是 **regress** 和 **anova** 用的常规平均数）之间的差异。以类似的工作方式，**qreg** 提供了第三种可能来检验中位数之间的差异。为了比较，下面来进行教师工资分析的分位数回归：

```
. qreg pay assoc full female femassoc femfull, nolog

Median regression                                    Number of obs =       226
  Raw sum of deviations   1738010 (about 37360)
  Min sum of deviations    798870                    Pseudo R2     =    0.5404

------------------------------------------------------------------------------
         pay |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       assoc |       -760   440.1693     -1.73   0.086    -1627.488    107.4881
        full |      10335   615.7735     16.78   0.000      9121.43    11548.57
      female |  -623.3333   365.1262     -1.71   0.089    -1342.926     96.2594
    femassoc |  -156.6667   440.1693     -0.36   0.722    -1024.155    710.8214
     femfull |  -691.6667   615.7735     -1.12   0.263    -1905.236    521.9031
       _cons |      38300   365.1262    104.90   0.000     37580.41    39019.59
------------------------------------------------------------------------------

. test assoc full

 ( 1)  assoc = 0.0
 ( 2)  full = 0.0

       F(  2,   220) =  208.94
            Prob > F =    0.0000
```

```
. test female

 ( 1)   female = 0.0

        F(  1,    220) =     2.91
             Prob > F =     0.0892

. test femassoc femfull

 ( 1)   femassoc = 0.0
 ( 2)   femfull = 0.0

        F(  2,    220) =     1.60
             Prob > F =     0.2039

. predict predpay3
(option xb assumed; fitted values)

. label variable predpay3 "Median predicted salary"

. table gender rank, contents(mean predpay3)

----------------------------------
Gender     |
(dummy     |     Academic rank
variable)  | Assist   Assoc    Full
-----------+----------------------
      Male |  28500   38320   49950
    Female |  28950   36760   47320
----------------------------------
```

由分位数回归得到的预测值与各交互分组中的中位工资数非常接近,因为我们可以直接核实:

```
. table gender rank, contents(median pay)

----------------------------------
Gender     |
(dummy     |     Academic rank
variable)  | Assist   Assoc    Full
-----------+----------------------
      Male |  28500   38320   49950
    Female |  28950   36590   46530
----------------------------------
```

于是,**qreg** 使我们像多因素方差分析或协方差分析那样来拟合模型,但是却通过 0.5 分位数或近似中位数的方式,而不是常规的平均数方式。在理论上,0.5 分位数和中位数是相同的。但是在实际中,分位数是用实际样本数值近似计算的,而当一个分组包括了偶数观测时,中位数却是通过位于最中间的两个值取平均值得到。所以,样本中的中位数和 0.5 分位数可能有点差别,但这种差别不至于影响到模型解释。

对 rreg 和 qreg 的更多应用

诊断统计与绘图(第 7 章)和非线性转换(第 8 章)扩展了稳健程序的用途,就像它们在常规回归时所做的那样。通过转换变量,**rreg** 或 **qreg** 也能拟合曲线回归模型。**rreg** 还能稳健地完成更简单的分析方法。要计算某一个变量平均数的 90% 置信区间,我们既可以键入通常置信区间的命令 **ci** :

```
. ci y, level(90)
```

也可以通过做没有 *x* 变量的回归来取得同样的平均数和置信区间：

```
. regress y, level(90)
```

同样，我们还能取得稳健平均数及其 90% 置信区间：

```
. rreg y, level(90)
```

qreg 也能以同样方式使用，但是要记住上节的提示，由 **qreg** 求出的 0.5 分位数可能与样本中位数有所不同。在以上这些命令中，选项 **level( )** 指定所要求的置信度。如果我们省略这个选项，**Stata** 就会自动显示 95% 置信区间。

要想比较两组的平均数，典型的作法是应用双样本 *t* 检验(**ttest**)或单因素方差分析(**oneway** 或 **anova**)。如前所示，我们可以用回归来完成等价的检验(得到相同的 *t* 和 *F* 统计量)。比如，将测量变量对代表两种类别的虚拟变量(这里称为 *group*)回归：

```
. regress y group
```

需要这种检验的稳健版本，可以键入以下命令：

```
. rreg y group
```

在默认状态，**qreg** 所完成的是中位数回归，但是它实际上是一个更综合的工具。它并不只限于中位数(0.5 分位数)，还能够对 *y* 的任意分位数来拟合线性模型。比如，以下命令就是分析 *y* 的第一四分位(0.25 分位数)是如何随 *x* 变化的。

```
. qreg y x, quant(.25)
```

假如误差方差相同，那么 0.25 和 0.75 分位数线的斜率应该大体相同。因此，**qreg** 可以用于检验异方差性问题或种种非线性问题。

方差的稳健估计—1

数据存在特异值倾向或非正态误差时，**rreg** 与 **qreg** 都比 OLS 方法(**regress** 和 **anova**)的效果好。然而，所有这些程序都共享同样的假定，即误差服从独立同分布。要是误差分布在不同 *x* 值上或不同案例之间有变化，那么由 **anova**、**regress**、**rreg** 以及 **qreg** 计算的标准误都可能会低估真正的样本与样本之间的变异，求出一个不切实际的狭窄的置信区间。

regress 和一些其他模型拟合命令(然而不包括 **rreg** 或 **qreg**)都有一个选项，可以不依赖于那些强硬的、有时是不合理的误差独立、同分布的假定，估计出标准误。这一选项应用了由 Huber、White 以及其他人分别独立推导出的一个方法，这一方法有时被称为方差的三明治估计(sandwich estimator)。一个人造数据(*robust2.dta*)提供了第一个例子。

当我们将 *y8* 对 *x* 回归，就会得到显著的正斜率。然而，散点图却表明了异方差性的存在(图 9.5)。回归线周围的变异随着 *x* 加大。因为误差并不是在所有 *x* 值上相同分布，由 **regress** 输出的标准误、置信区间和统计检验都是靠不住的。**rreg** 或 **qreg** 也同样面对这个问题。

```
Contains data from C:\data\robust2.dta
  obs:           500                          Robust regression examples 2
                                                (artificial data)
 vars:            12                          17 Jul 2005 09:03
 size:        24 500  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
x               float  %9.0g                  Standard normal x
e5              float  %9.0g                  Standard normal errors
y5              float  %9.0g                  y5 = 10 + 2*x + e5 (normal
                                                i.i.d. errors)
e6              float  %9.0g                  Contaminated normal errors:
                                                95% N(0,1), 5%(N(0,10)
y6              float  %9.0g                  y6 = 10 + 2*x + e6
                                                (Contaminated normal errors)
e7              float  %9.0g                  Centered chi-square(1) errors
y7              float  %9.0g                  y7 = 10 + 2*x + e7 (skewed
                                                errors)
e8              float  %9.0g                  Normal errors, variance
                                                increases with x
y8              float  %9.0g                  y8 = 10 + 2*x + e8
                                                (heteroskedasticity)
group           byte   %9.0g
e9              float  %9.0g                  Normal errors, variance
                                                increases with x, mean &
                                                variance increase with cluster
y9              float  %9.0g                  y9 = 10 + 2*x + e9
                                                (heteroskedasticity &
                                                correlated errors)
-------------------------------------------------------------------------------
Sorted by:
```

. regress *y8 x*

```
      Source |       SS       df       MS              Number of obs =     500
-------------+------------------------------           F(  1,   498) =  133.96
       Model |  1607.35658     1  1607.35658           Prob > F      =  0.0000
    Residual |  5975.19162   498  11.9983767           R-squared     =  0.2120
-------------+------------------------------           Adj R-squared =  0.2104
       Total |   7582.5482   499  15.1954874           Root MSE      =  3.4639

------------------------------------------------------------------------------
          y8 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.819032   .1571612    11.57   0.000     1.510251    2.127813
       _cons |   10.06642    .154919    64.98   0.000     9.762047     10.3708
------------------------------------------------------------------------------
```

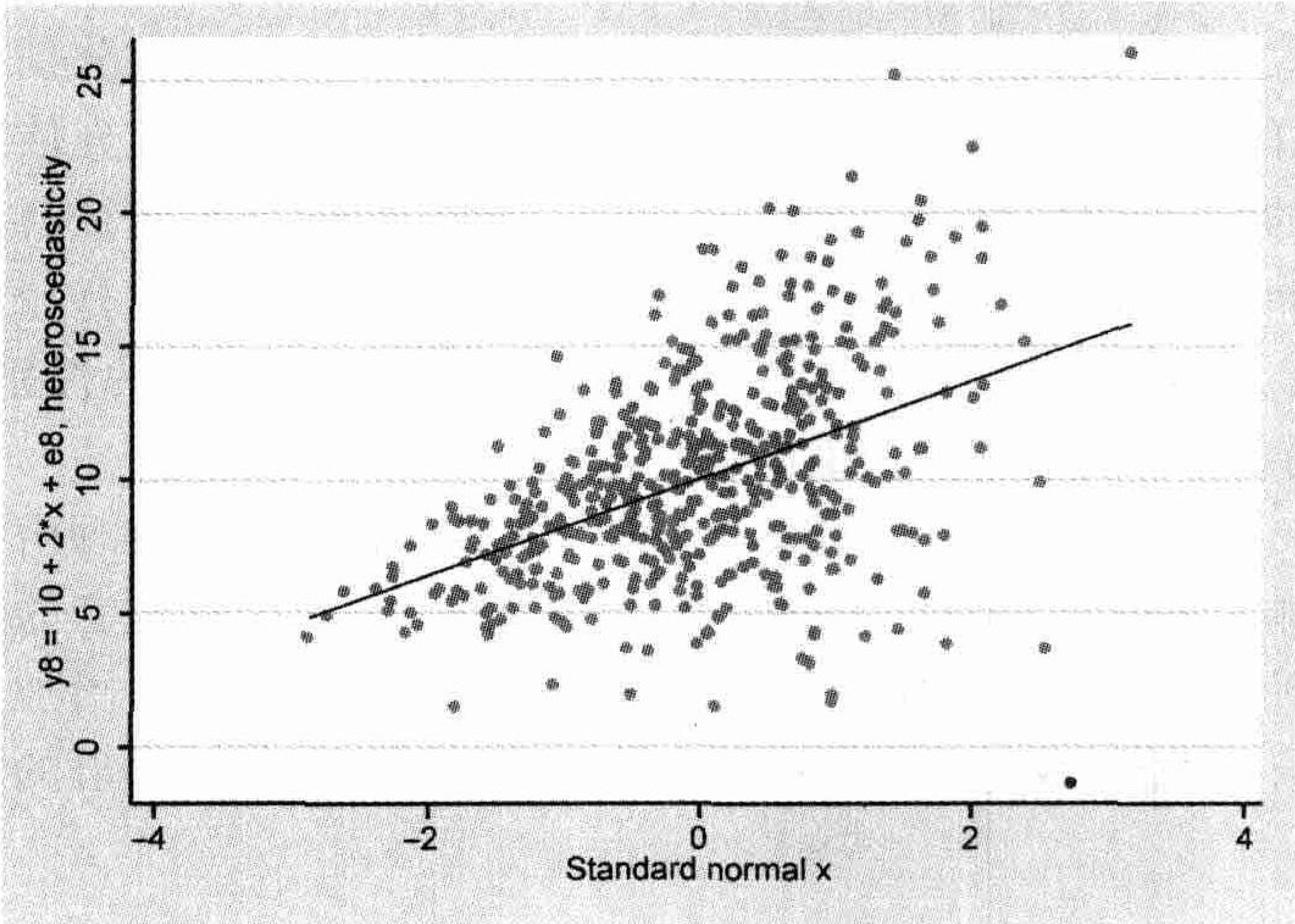


图 9.5

对这一 OLS 回归的更可信的标准误和置信区间可以用 **robust** 选项来得到:

```
. regress y8 x, robust

Regression with robust standard errors                 Number of obs =     500
                                                       F(  1,   498) =   83.80
                                                       Prob > F      =  0.0000
                                                       R-squared     =  0.2120
                                                       Root MSE      =  3.4639

------------------------------------------------------------------------------
             |               Robust
          y8 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   1.819032   .1987122     9.15   0.000     1.428614    2.209449
       _cons |   10.06642   .1561846    64.45   0.000     9.759561    10.37328
------------------------------------------------------------------------------
```

尽管拟合的模型未变,斜率的稳健标准误比前面的非稳健估计大了 27% (0.199 对比 0.157)。用了 **robust** 选项后,回归不再输出通常的方差分析表,因为这些已经不再具有通常的解释意义。

这些稳健标准误估计背后的原理在《用户指南》中有所解释。简而言之,我们放弃了估计下列模型的真实总体参数(β)的经典目标:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

作为较低目标,我们追求单纯地估计出 *b* 系数在样本与样本之间变异性。如果我们可以抽出许多随机样本并且重复地应用 OLS 估计,最后可以计算出以下模型中的 *b* 值:

$$y_i = b_0 + b_1 x_i + e_i$$

我们并不假定这些 *b* 估计将收敛于某个"真实"的总体参数。所以,用稳健标准误形成的置信区间并不适用那种以特定概率(通过重复抽样)将总体参数包括在内的经典解释意义。相反,稳健置信区间有特定概率(通过重复抽样)将 *b* 包括在内,而 *b* 定义为样本 *b* 估计的收敛值。于是,我们有得有失,一方面放松了误差同分布的假定,另一方面只能满足于一个不那么深刻的结论。

方差的稳健估计—2

另一种稳健方差的选项 **cluster** 使我们能以一种有限的方式放松误差独立的假定,这就是当误差在数据分组或类别之内相关的时候。数据 *attract.dta* 可以用于示范这种情况,它描述了一个大学生的社会实验。在这个实验中,51 名大学生被分别要求对一些不认识的男人和女人的照片进行魅力打分,尺度为从 1 到 10。这种打分依次由每个学生重复,给他们同样的照片但是随机改变照片的顺序。这个实验在晚间社会活动时进行了 4 次。变量 *ratemale* 是每个参加者在一次试验中对所有男人照片打分的平均数,*ratefem* 是对所有女人照片打分的平均数。*gender* 是打分者自己的性别,而 *bac* 是当时用呼吸分析仪测出打分者当时的血液酒精含量。

尽管这个数据包含了 204 个观测,但是它们只代表 51 个参加者。所以有理由认为扰动项(打分中未测量的影响)在每个人的几次重复试验中相关。将每个参加者的四次打分视为一个类群,应该有助于得到更符合实际的标准误估计。在回归命令中附加上选项 **cluster(*id*)**,如下所示,便能够取得贯穿由 *id*(即个人识别码)定义的类群的稳健标准误。

```
Contains data from C:\data\attract.dta
  obs:           204                          Perceived attractiveness and
                                                drinking (D. C. Hamilton 2003)
 vars:             8                          18 Jul 2005 17:27
 size:         5 508 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              byte   %9.0g                  Participant number
gender          byte   %9.0g       sex        Participant gender (female)
bac             float  %9.0g                  Blood alchohol content
genbac          float  %9.0g                  gender*bac interaction
relstat         byte   %9.0g       rel        Relationship status (single)
drinkfrq        float  %9.0g                  Days drinking in previous week
ratefem         float  %9.0g                  Rated attractiveness of females
ratemale        float  %9.0g                  Rated attractiveness of males
-------------------------------------------------------------------------------
Sorted by:  id
```

```
. regress ratefem bac gender genbac, cluster(id)
```

```
Regression with robust standard errors                 Number of obs =     204
                                                       F(  3,    50) =    7.75
                                                       Prob > F      =  0.0002
                                                       R-squared     =  0.1264
Number of clusters (id) = 51                           Root MSE      =  1.1219

------------------------------------------------------------------------------
             |               Robust
     ratefem |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         bac |   2.896741   .8543378     3.39   0.001     1.180753    4.612729
      gender |  -.7299888   .3383096    -2.16   0.036    -1.409504   -.0504741
      genbac |   .2080538   1.708146     0.12   0.904    -3.222859    3.638967
       _cons |   6.486767    .229689    28.24   0.000     6.025423     6.94811
------------------------------------------------------------------------------
```

血液酒精度(*bac*)存在显著的正效应:当 *bac* 升高时,对女人照片的魅力打分也会提高。性别(女性)存在负效应:相比男学生,女学生倾向于对女人照片的魅力分打得低(约低 0.73 分)。性别与血液酒精度的交互项 *genbac* 影响很小(0.21)。这一截距和斜率都受虚拟变量影响的回归模型的通用表达式为:

预测值: $ratefem = 6.49 + 2.90bac - 0.73gender + 0.21genbac$

亦可简化为对男生的预测($gender = 0$),有:

预测值: $ratefem = 6.49 + 2.90bac - (0.73 \times 0) + (0.21 \times 0 \times bac)$

$= 6.49 + 2.90bac$

还可化为对女生的预测($gender = 1$),有:

预测值: $ratefem = 6.49 + 2.90bac - (0.73 \times 1) + (0.21 \times 1 \times bac)$

$= 6.49 + 2.90bac - 0.73 + 0.21 \times bac$

$= 5.76 + 3.11bac$

我们看到酒精对男生和女生的影响存在差别,对男生的影响(2.90)和对女生的影响(3.11)之间的差别来自于交互项系数 0.21。

对男人照片的魅力打分受到血液酒精度的正影响。性别对男人照片打分有很大影响:女生倾向于给男人照片打出比男生高得多的分。在为男人照片打分时,性别和酒精度交互项的影响巨大(-4.36),尽管它并未达到 0.05 显著水平。

```
. regress ratemal bac gender genbac, cluster(id)
```

```
Regression with robust standard errors                 Number of obs =     201
                                                       F(  3,     50) =   10.96
                                                       Prob > F      =  0.0000
                                                       R-squared     =  0.3516
Number of clusters (id) = 51                           Root MSE      =  1.3931

------------------------------------------------------------------------------
             |               Robust
   ratemale  |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         bac |   4.246042   2.261792     1.88   0.066    -.2969004    8.788985
      gender |   2.443216   .4529047     5.39   0.000      1.53353    3.352902
      genbac |  -4.364301   3.573689    -1.22   0.228    -11.54227    2.813663
       _cons |   3.628043   .2504253    14.49   0.000     3.125049    4.131037
------------------------------------------------------------------------------
```

男生对男人照片打分的回归方程可化为：

预测值： $ratemale = 3.63 + 4.25 bac + (2.44 \times 0) - (4.36 \times 0 \times bac)$

$= 3.63 + 4.25 bac$

而女生对男人照片打分的回归方程可化为：

预测值： $ratemale = 3.63 + 4.25 bac + (2.44 \times 1) - (4.36 \times 1 \times bac)$

$= 6.07 - 0.11 bac$

酒精对男生和女生的影响有很大差别，对男生的的影响较大(4.25)，而对女生没什么影响(-0.11)，这种差距正好等于交互效应系数-4.36。在这个样本中，当打分者的血液酒精度提高时，男生对男人照片打的分迅速飙升，而女生在对男人照片打分时则很稳定。

图9.6将这些结果绘在一起。我们可以看到，酒精度对打分的正影响贯穿于所有子图中，只是当女生为男人打分时除外。这些图形还显示出其他的性别差异，包括男生有较高的血液酒精度测量值。

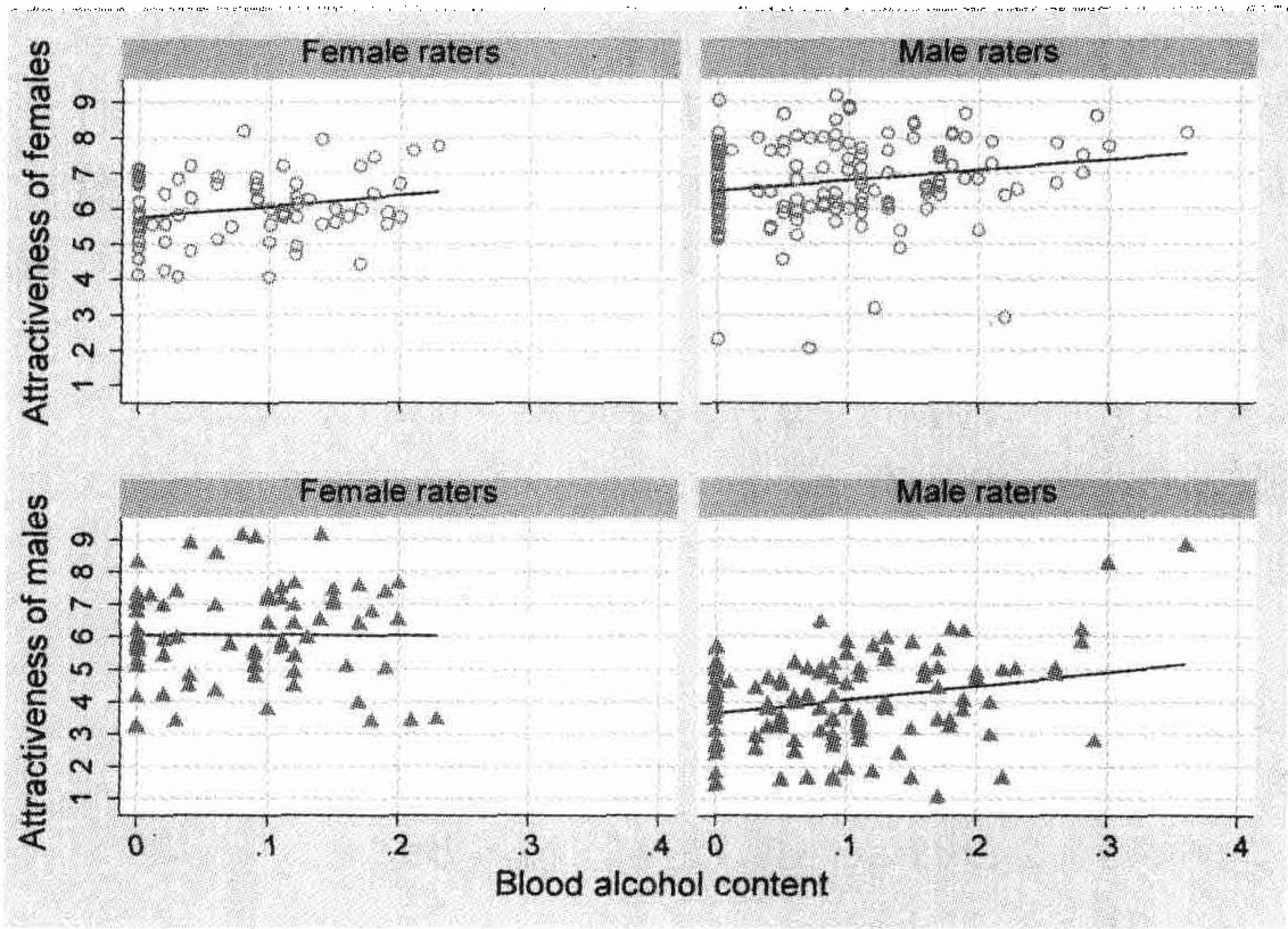


图 9.6

用 **regress** 加 **robust** 选项可以进行 OLS 回归并估计出稳健标准误，不要将这种方法与用 **rreg** 所做的稳健回归相混淆。尽管这两者的称呼很类似，但它们是两种并无联系的程序，而且解决的是不同的问题。

10 logistic 回归

第 5 章至第 9 章介绍的回归和方差分析方法都要求因变量(即 y 变量)是测量水平的。Stata 还提供了一整套对分类变量、序次变量以及删截的因变量的建模方法。下面列出一些有关命令的清单(如需更详细的说明,可键入 **help** 命令查询):

binreg 二项回归(binomial regression),属一般化线性模型(generalized linear models)。

blogit 对分组(或分块,blocked)数据的 logit 估计方法(logit estimation)。

bprobit 对分组(或分块)数据的 probit 估计方法(probit estimation)。

clogit 条件固定效应的 logistic 回归(conditional fixed-effects logistic regression)。

cloglog 补充双对数估计方法(complementary log-log estimation)。

cnreg 删截正态回归(censored-normal regression),在假定 y 服从高斯分布但在某一点(案例之间不同)被删截时使用。

constraint 用于定义、列出和取消线性约束条件。

dprobit 提供概率变化(而不是系数)的 probit 回归。

glm 一般化线性模型(generalized linear models),包括了 logistic、probit 或补充双对数等连接方式,允许反应变量为二分变量或与分组成比例的变量。

glogit 对分组数据的 logit 回归(logit regression for grouped data)。

gprobit 对分组数据的 probit 回归(probit regression for grouped data)。

heckprob 带选择的 probit 估计(probit estimation with selection)。

hetprob 异方差的 probit 估计(heteroskedastic probit estimation)。

intreg 区间回归(interval regression),其中 y 可以为点数据、区间数据以及左删截或右删截数据。

logistic logistic 回归(logistic regression),输出优势比(odds ratios)结果。

logit logistic 回归,与 **logistic** 类似,但输出回归系数。

mlogit 多项 logistic 回归(multinomial logistic regression),用于多分类的 y 变量。

nlogit 嵌套的 logit 估计(nested logit estimation)。

ologit 序次 logistic 回归(logistic regression with ordinal y variable),用于序次的 y 变量。

oprobit 序次 probit 回归(probit regression with ordinal y variable)。

probit probit 回归(probit regression),用于二分类的 y 变量。

rologit 等级排序的 logit 模型(rank-ordered logit model),用于等级因变量(也称 Plackett-Luce 模型、分解 logit 模型(exploded logit model)或基于选择的联合分析(choice-based conjoint analysis))。

scobit 偏态 probit 估计(skewed probit estimation)。

svy: logit 调查数据版的 logistic 回归,用于复杂调查数据。在调查数据版(**svy**)中还有许多其他分类变量模型的命令。

tobit tobit 回归(tobit regression),在假定 y 服从高斯分布但在已知固定点被删截时使用(这类的一般化模型参见 **help cnreg**)。

xtcloglog 随机效应与总体平均的补充双对数模型(Random-effects and population-averaged cloglog models)。还有面板(panel)数据版(**xt**)的 **logit** 和 **probit** 模型,以及总体平均的一般化线性模型,参见 **help xtgee**。

在所有模型拟合命令执行之后,都可以用 **predict** 命令来计算预测值或概率。**predict** 命令还可以取得适当的诊断统计,比如,那些由 Hosmer 和 Lemeshow (2000)所描述的关于 logistic 回归的诊断统计。特定的 **predict** 选项的功能依赖于刚刚拟合的模型类型。还有一个不同的拟合后续命令 **predictnl** 能取得非线性预测值及其相应置信区间(参见 **help predictnl**)。

下一节将示范其中几个命令的使用。对分类因变量建模的大多数方法可以在以下菜单中找到:

Statistics-Binary outcomes 二分类结果

Statistics-Ordinal outcomes 序次结果

Statistics-Categorical outcomes 分类结果

Statistics-Generalized linear models (GLM) 一般化线性模型

Statistics-Longitudinal/panel data 纵贯及面板数据

Statistics-Linear regression and related-Censored regression 删截回归

在以下命令示范这一节之后,本章的其余部分将集中介绍 logit 或 logistic 回归这一类重要方法。我们将逐步介绍对二分因变量、序次因变量和多分类因变量的 logit 模型。

命令示范

. logistic *y x1 x2 x3*

对{0,1}编码变量 *y* 执行 logistic 回归,自变量为 *x1*、*x2*、*x3*。自变量的影响是以优势比(odds ratio)形式输出的。与其紧密相连的命令:

. logit *y x1 x2 x3*

执行的是同样的分析,只是输出的是 logit 模型的回归系数。**logistic** 和 **logit** 所拟合的是同一个模型,因此后续得到的预测值或诊断检验都是相同的。

. lfit

提供对所拟合的 logistic 模型的皮尔森卡方拟合优度检验(Pearson chi-squared goodness-of-fit test):用 *y* = 1 情况的观测频数对比期望频数,按协变量(covariate,即 *x* 变量)的模式定义交互单元。当 *x* 的模式数目很大时,我们可能想要根据其估计概率将它们分组。命令 **lfit, group(10)** 将按 10 个大致等规模分组进行检验。

. lstat

提供分类统计和分类表。当分析点为分类情况时,命令 **lstat**、**lroc**、**lsens**(参见下面)都尤其重要。这些命令都是参照于前面刚刚拟合的 **logistic** 模型。

. lroc

画出接收器运行特征(receiver operating characteristic, ROC)的曲线制图,并计算这一曲线下的面积。

. lsens

提供敏感性(sensitivity)和特异性(specificity)分别对概率分割点(probability cutoff)的制图。

. predict *phat*

创建一个新变量(这里任意命名为 *phat*),等于最近一次 **logistic** 模型基础上 *y* = 1 的预测概率。

. predict *dX2*, dx2

形成一个新变量(任意命名为 *dX2*),记录诊断性统计量,测量相对最近一次 **logistic** 分析的皮尔森卡方变化量。

. mlogit *y x1 x2 x3*, base(3) rrr nolog

将多分类变量 *y* 与 3 个 *x* 变量做多项 logistic 回归。选项 **base(3)** 指定 *y* = 3 这类作为比较的基准类别;选项 **rrr** 要求用相对风险比(relative risk ratios)输出代替回归系数输出;选项 **nolog** 取消了迭代过程中对数似然值的显示。

. predict *P2*, outcome(2)

新建一个变量(任意命名为 *P2*),基于最近一次 **mlogit** 分析,记录 *y* = 2 的预测概率。

. glm *success x1 x2 x3*, family(binomial *trials*) eform

采用一般化线性模型来执行 logistic 回归,并且应用的是列表数据而不是个体观测数

据。变量 *success* 是所关注结果发生的频数，*trial* 是对自变量 *x1*、*x2*、*x3* 每种组合所含的频数。也就是说，*success* / *trial* 就是某一结果（比如，患者痊愈）发生次数所占的比例。选项 **eform** 要求输出优势比（指数幂形式），而不是 logit 系数。

```
. cnreg y x1 x2 x3, censored(cen)
```

将测量变量 *y* 与 3 个 *x* 自变量做删截正态回归。如果一个观测案例的 *y* 真值由于左删截或右删截而未知，在这一回归中它就由离其删截时最近的一个 *y* 值来取代。删截变量 *cen* 是一个编码为{ -1,0,1 }的标识(indicator)，编码值分别表示这个观测的 *y* 值是左删截、无删截、还是右删截。

航天飞机数据

本章的主要例子是美国航天飞机的前 25 次飞行的数据 *shuttle.dta*。这些数据包含的证据表明，如果这些数据早些得到适当的分析，就能说服美国航天署官员在 1985 年停止挑战者号的最后一次致命的飞行（即它的第 25 次航天飞行，派遣号为 STS 51-L）。这些数据来自于总统委员会对航天飞机挑战者号事故的报告(Report of the Presidential Commission on the Space Shuttle Challenger Accident 1986)，以及 Tufte 的著作(1997)。Tufte 的书中包括了对数据与分析方面的卓越讨论。他关于航天飞行细节的评论也作为字符串变量包括在这些数据中。

```
Contains data from C:\data\shuttle.dta
  obs:            25                          First 25 space shuttle flights
 vars:             8                          20 Jul 2005 10:40
 size:         1 675 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
flight          byte   %8.0g       flbl       Flight
month           byte   %8.0g                  Month of launch
day             byte   %8.0g                  Day of launch
year            int    %8.0g                  Year of launch
distress        byte   %8.0g       dlbl       Thermal distress incidents
temp            byte   %8.0g                  Joint temperature, degrees F
damage          byte   %9.0g                  Damage severity index (Tufte
                                                1997)
comments        str55  %55s                   Comments (Tufte 1997)
-------------------------------------------------------------------------------
Sorted by:
```

本章检查了 *shuttle.dta* 数据中的 3 个变量：

distress “热损事件”(thermal distress incidents)的数量，这些事件是因为热气泄漏或烧坏了这次航行助推火箭的结点密封。助推结点密封的烧穿使挑战者号陷入灾难。许多以前的航行也经历过不太严重的损坏，所以早就知道结点密封是危险的可能来源。

temp 在发射时间对结点的计算温度，以华氏度为单位。温度在很大程度上受天气影响。橡胶的 O 形环密封的助推火箭结点在很冷时会变得僵硬。

date 日期，测量方法是从 1960 年 1 月 1 日（一个指定的起始点）起的消逝天数。*date* 是用 **mdy** 函数由发射的月、日、年(month-day-year)转换成的消逝天数（参见 **help dates**）：

发射日期很重要，因为航天项目进程中的几个变化可能造成较大风险。助推火箭外层很薄以减少重量、增加有效载荷，于是结点密封就得能经受高压测试。此外，航天飞机回收再用也导致其各部分的老化。所以我们也许会问，助推结点损坏（一处或多处受损

事件)的可能性是否随发射日期而增加?

```
. list flight-temp, sepby(year)

     +-------------------------------------------------------------+
     |    flight   month   day   year   date   distress   temp |
     |-------------------------------------------------------------|
  1. |     STS-1       4    12   1981   7772       none     66 |
  2. |     STS-2      11    12   1981   7986     1 or 2     70 |
     |-------------------------------------------------------------|
  3. |     STS-3       3    22   1982   8116       none     69 |
  4. |     STS-4       6    27   1982   8213          .     80 |
  5. |     STS-5      11    11   1982   8350       none     68 |
     |-------------------------------------------------------------|
  6. |     STS-6       4     4   1983   8494     1 or 2     67 |
  7. |     STS-7       6    18   1983   8569       none     72 |
  8. |     STS-8       8    30   1983   8642       none     73 |
  9. |     STS-9      11    28   1983   8732       none     70 |
     |-------------------------------------------------------------|
 10. | STS_41-B       2     3   1984   8799     1 or 2     57 |
 11. | STS_41-C       4     6   1984   8862     3 plus     63 |
 12. | STS_41-D       8    30   1984   9008     3 plus     70 |
 13. | STS_41-G      10     5   1984   9044       none     78 |
 14. | STS_51-A      11     8   1984   9078       none     67 |
     |-------------------------------------------------------------|
 15. | STS_51-C       1    24   1985   9155     3 plus     53 |
 16. | STS_51-D       4    12   1985   9233     3 plus     67 |
 17. | STS_51-B       4    29   1985   9250     3 plus     75 |
 18. | STS_51-G       6    17   1985   9299     3 plus     70 |
 19. | STS_51-F       7    29   1985   9341     1 or 2     81 |
 20. | STS_51-I       8    27   1985   9370     1 or 2     76 |
 21. | STS_51-J      10     3   1985   9407       none     79 |
 22. | STS_61-A      10    30   1985   9434     3 plus     75 |
 23. | STS_61-B      11    26   1985   9461     1 or 2     76 |
     |-------------------------------------------------------------|
 24. | STS_61-C       1    12   1986   9508     3 plus     58 |
 25. | STS_51-L       1    28   1986   9524          .     31 |
     +-------------------------------------------------------------+

. generate date = mdy(month, day, year)
. label variable date "Date (days since 1/1/60)"
```

distress 是个有标签的数字编码变量:

```
. tabulate distress

Thermal      |
distress     |
incidents    |      Freq.     Percent        Cum.
-------------+-----------------------------------
        none |          9       39.13       39.13
      1 or 2 |          6       26.09       65.22
      3 plus |          8       34.78      100.00
-------------+-----------------------------------
       Total |         23      100.00
```

通常,**tabulate** 会显示标签,但是选项 **nolabel** 揭示出,数字码 0 =“无”,1 =“1 次或 2 次”,2 =“3 次及以上”。

```
. tabulate distress, nolabel

Thermal      |
distress     |
incidents    |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |          9       39.13       39.13
           1 |          6       26.09       65.22
           2 |          8       34.78      100.00
-------------+-----------------------------------
       Total |         23      100.00
```

我们可以用这些编码来新建一个虚拟变量 *any*,用 0 代表无损坏,1 代表有 1 处或更多损坏事件:

```
. generate any = distress
(2 missing values generated)

. replace any = 1 if distress == 2
(8 real changes made)

. label variable any "Any thermal distress"
```

要想看看都做了些什么,

```
. tabulate distress any

Thermal     | Any thermal distress
distress    |
incidents   |         0           1 |     Total
------------+-----------------------+----------
       none |         9           0 |         9
     1 or 2 |         0           6 |         6
     3 plus |         0           8 |         8
------------+-----------------------+----------
      Total |         9          14 |        23
```

logistic 回归对 *any* 这种{0,1}二分因变量建模,对一个或更多的 x 变量回归。命令 **logit** 与 **regress** 及大多数其他模型拟合命令的形式类似,都是将因变量列为第一个变量。

```
. logit any date, coef

Iteration 0:   log likelihood = -15.394543
Iteration 1:   log likelihood =  -13.01923
Iteration 2:   log likelihood = -12.991146
Iteration 3:   log likelihood = -12.991096
Logit estimates                                   Number of obs   =         23
                                                  LR chi2(1)      =       4.81
                                                  Prob > chi2     =     0.0283
Log likelihood = -12.991096                       Pseudo R2       =     0.1561

------------------------------------------------------------------------------
         any |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        date |   .0020907   .0010703     1.95   0.051    -6.93e-06    .0041884
       _cons |  -18.13116   9.517217    -1.91   0.057    -36.78456    .5222396
------------------------------------------------------------------------------
```

logit 迭代估计程序将使对数似然函数最大化,正如输出的上部所示。在迭代开始(Iteration 0)时,对数似然值(log likelihood)反映了模型中只有截距时的拟合状况。而最后一个对数似然值则表示最终模型的拟合状况:

$$L = -18.131\,16 + 0.002\,090\,7 date \qquad [10.1]$$

其中 L 代表损坏发生事件的 logit 预测值,或称为对数发生比(log odds):

$$L = \ln[P(any = 1) / P(any = 0)] \qquad [10.2]$$

输出表的右上部有模型整体卡方(χ^2)检验(虚无假设即模型中除了截距外的所有系数都等于 0)的结果[10]:

$$\chi^2 = -2(\ln \mathbb{L}_i - \ln \mathbb{L}_f) \qquad [10.3]$$

其中 $\ln \mathbb{L}_i$ 初始或迭代 0 时(即截距模型)的对数似然值,而 $\ln \mathbb{L}_f$ 为最终迭代的对数似然

[10]【译注:该输出表中将整体似然比卡方标注为 LR chi2(1),括号之中数字为卡方分布的自由度。】

值。有：

$$\chi^2 = -2[-15.394\ 543-(-12.991\ 096)] \\ = 4.81$$

这个较大的卡方值在对应1个自由度（即初始模型与最终模型的复杂性差别）时的概率已经足够小（0.028 3），以致本例的虚无假设遭到拒绝。结果表明，*date* 的确有显著的影响。

logit 结果输出中还提供了渐近 *z*（标准正态）检验，这种检验不太准确，但是很方便。在只有一个自变量时，对这个自变量的 *z* 检验与模型整体卡方检验是等价的（检验的是同一个假设），这与在简单 OLS 回归时的 *t* 检验和 *F* 检验的关系很类似。与 OLS 不同的是，logit 的 *z* 近似检验和卡方检验有时并不吻合（这里就不同）；而卡方检验具有更高的有效性。

与 Stata 其他最大似然估计程序一样，**logit** 的输出中提供了一个伪（pseudo）R^2：

$$\text{pserdo } R^2 = 1 - \ln \mathbb{L}_f / \ln \mathbb{L}_i \qquad [10.4]$$

对于这个例子有，

$$\text{pseudo } R^2 = 1-(-12.991\ 096)/(-15.394\ 543) \\ = 0.156\ 1$$

尽管它提供了一个便捷方式来描述或比较对同一因变量的不同模型的拟合状况，伪 R^2 统计量缺乏像 OLS 回归中真 R^2 那样的方差解释意义。

在执行 **logit** 以后，**predict** 命令（不加任何选项）将取得预测概率：

$$Phat = 1/(1+e^{-L}) \qquad [10.5]$$

按 *date* 作图，从图 10.1 中可以看到，这些概率呈一种 *S* 状的 logistic 曲线。

```
. predict Phat
. label variable Phat "Predicted P(distress >= 1)"
. graph twoway connected Phat date, sort
```

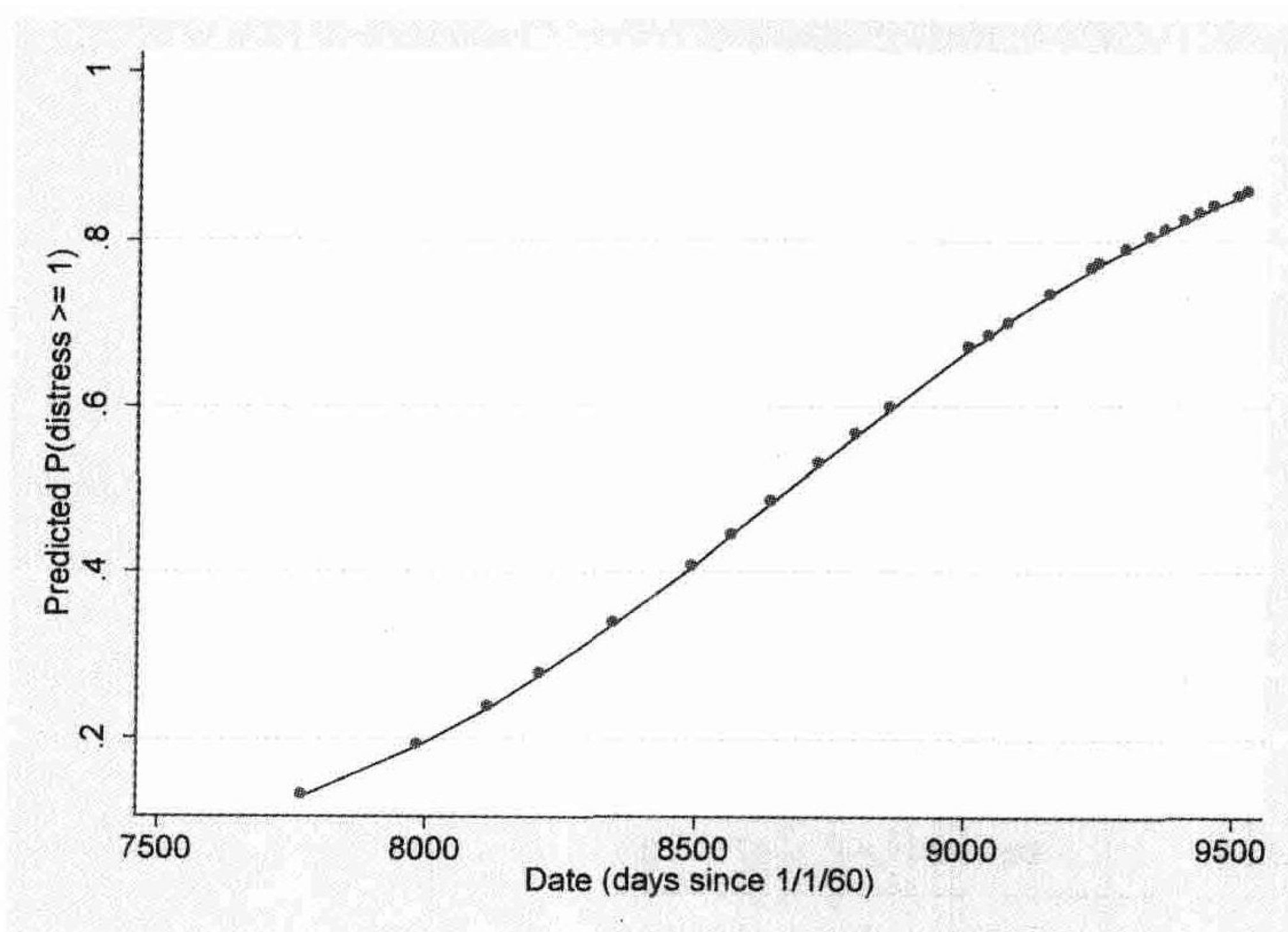


图 10.1

logit 提供的系数（0.002 090 7）描述了 *date* 对热损事件发生的 logit 转换值即对数发生比的影响。每一天的增加都使热损事件对数发生比预测值提高 0.002 090 7。换句话说，每一天的增加将使热损发生比的预测值是前一天的 $e^{0.002\ 090\ 7}=1.002\ 092\ 9$ 倍，那

么每100天后热损发生比的变化倍数为($e^{0.002\,090\,7}$)^{100} =1.23 倍。(这里 e≈2.718 28,即自然对数的底。) Stata 也可以进行这些计算,只需要调用估计之后内存中的 _b[*varname*]系数:

```
. display exp(_b[date])
1.0020929

. display exp(_b[date])^100
1.2325359
```

或者,我们也可以在 **logit** 命令中简单加上选项 **or**(代表 odds ratio)也行。再一种方法是下一节要讲的用 **logistic** 命令来取得发生比。**logistic** 命令与 **logit** 在拟合模型上完全一样,但是其默认输出表中提供的就是发生比,而不是系数。

使用 logistic 回归

这里我们用 **logistic** 命令来做前面用 **logit** 估计的同样回归模型:

```
.  logistic any date

Logit estimates                                   Number of obs   =         23
                                                  LR chi2(1)      =       4.81
                                                  Prob > chi2     =     0.0283
Log likelihood = -12.991096                       Pseudo R2       =     0.1561

------------------------------------------------------------------------------
         any | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        date |   1.002093   .0010725     1.95   0.051     .9999931    1.004197
------------------------------------------------------------------------------
```

注意,这里的对数似然值和卡方统计量都与前面取得的相同。**logistic** 不再提供系数(b),而是提供优势比(Odds Ratio,即 e^b)。它的意义是,自变量每增加一个单位时,事件($y=1$)的发生比的变化倍数(如有其他自变量,则以其他自变量保持不变为条件)。

在拟合一个模型以后,我们能通过命令来取得分类表及其有关统计:

```
. lstat

Logistic model for any

              -------- True --------
Classified |         D            ~D  |      Total
-----------+--------------------------+-----------
     +     |        12             4  |         16
     -     |         2             5  |          7
-----------+--------------------------+-----------
   Total   |        14             9  |         23

Classified + if predicted Pr(D) >= .5
True D defined as any != 0
--------------------------------------------------
Sensitivity                     Pr( +| D)   85.71%
Specificity                     Pr( -|~D)   55.56%
Positive predictive value       Pr( D| +)   75.00%
Negative predictive value       Pr(~D| -)   71.43%
--------------------------------------------------
False + rate for true ~D        Pr( +|~D)   44.44%
False - rate for true D         Pr( -| D)   14.29%
False + rate for classified +   Pr(~D| +)   25.00%
False - rate for classified -   Pr( D| -)   28.57%
--------------------------------------------------
Correctly classified                        73.91%
--------------------------------------------------
```

按照默认状况,**lstat** 应用 0.5 的概率作为分割点(尽管我们可以加入 **cutoff( )** 选项来改变这一设置)。分类表中的几种符号有以下含义:

D　一个观测中所关注的事件确实发生(即 $y=1$)。在本例中,D 表示热损发生了。

~D　一个观测中所关注的事件没有发生(即 $y=0$)。在本例中,~D 表示本次航行没发生热损。

+　模型预测概率值大于等于分割点。由于我们使用了默认的分割点,这里的 + 表示模型预测的发生热损概率为 0.5 或更高。

-　预测概率值小于分割点。这里,- 表示预测的热损概率小于 0.5。

于是,按照模型预测热损概率值至少在 0.5 以上的标准,有 12 次航行的分类是准确的,即损坏的确实际上发生了。另外的 5 次航行,模型预测概率小于 0.5,并且损坏并没有发生。因此总的分类正确率为 12 +5 =17 再除以 23,即 73.91% 。这个表还提供了一些条件概率,比如,敏感性(sensitivity)即在热损发生案例中预测概率大于等于 0.5 的案例所占的百分比(14 次中的 12 次,即 85.71%)。

在执行 **logistic** 或 **logit** 后,后续命令 **predict** 可以计算各种预测和诊断用的统计指标。有关诊断指标的讨论请参见 Hosmer 和 Lemeshow(2000)。

predict *newvar*	预测 $y=1$ 的概率
predict *newvar*, xb	线性预测(即 $y=1$ 的对数发生比)
predict *newvar*, stdp	线性预测的标准误
predict *newvar*, dbeta	ΔB 影响统计量,类似于 Cook 的 D 统计量
predict *newvar*, deviance	对第 j 个 x 模式的偏差度残差 d_j
predict *newvar*, dx2	皮尔森卡方变化量,记为 $\Delta\chi^2$ 或 $\Delta\chi^2_P$
predict *newvar*, ddeviance	偏差度卡方变化量,记为 ΔD 或 $\Delta\chi^2_D$
predict *newvar*, hat	对第 j 个 x 模式的杠杆作用 h_j
predict *newvar*, number	第 j 个 x 模式的分配数,$j=1,2,3,\cdots,J$
predict *newvar*, resid	第 j 个 x 模式的皮尔森残差 r_j
predict *newvar*, rstandard	标准化皮尔森残差

用 **dbeta** 、**dx2** 、**ddeviance** 和 **hat** 等选项取得的统计量并不是测量个别观测影响的,这与常规回归的对应部分一样。这些统计量测量的是"协变量模式"(covariate patterns)的影响,即如果将有特定 x 组合值的所有案例排除后的影响。(有关详细讨论参见 Hosmer 和 Lemeshow(2000)。本章的后面将会示范这些统计量的使用)

是否助推结点温度也会影响损坏事件的概率呢?我们可以将 *temp* 作为第二个预测变量加入模型来对此进行研究。

```
. logistic any date temp

Logit estimates                                   Number of obs   =         23
                                                  LR chi2(2)      =       8.09
                                                  Prob > chi2     =     0.0175
Log likelihood = -11.350748                       Pseudo R2       =     0.2627

------------------------------------------------------------------------------
         any | Odds Ratio   Std. Err.      z     P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        date |    1.00297   .0013675     2.17   0.030     1.000293    1.005653
        temp |   .8408309   .0987887    -1.48   0.140     .6678848    1.058561
------------------------------------------------------------------------------
```

分类表显示出,将温度作为自变量可以将我们的分类正确率提高到78.26%。

```
. lstat

Logistic model for any

              -------- True --------
Classified |         D            ~D  |      Total
-----------+--------------------------+-----------
     +     |        12             3  |         15
     -     |         2             6  |          8
-----------+--------------------------+-----------
   Total   |        14             9  |         23

Classified + if predicted Pr(D) >= .5
True D defined as any != 0
--------------------------------------------------
Sensitivity                     Pr( +| D)   85.71%
Specificity                     Pr( -|~D)   66.67%
Positive predictive value       Pr( D| +)   80.00%
Negative predictive value       Pr(~D| -)   75.00%
--------------------------------------------------
False + rate for true ~D        Pr( +|~D)   33.33%
False - rate for true D         Pr( -| D)   14.29%
False + rate for classified +   Pr(~D| +)   20.00%
False - rate for classified -   Pr( D| -)   25.00%
--------------------------------------------------
Correctly classified                        78.26%
--------------------------------------------------
```

根据这个拟合模型,结点温度每1度增量将使助推结点损坏发生比乘以0.84(换句话说,温度每提高1度减少损坏发生比16%)。尽管这种影响看起来很大,值得关注,近似 z 检验却表明它的统计性并不显著($z=-1.476$, $P=0.140$)。然而,应用似然比卡方检验才更有确定性。命令 **lrtest** 根据最大似然估计值来比较嵌套的模型。首先,先要估计出包括所有自变量的完整(full)模型,就像前面用 **logistic *any date temp*** 命令那样。然后,再键入 **estimates store** 命令,并指定一个变量名(如 *full*)来识别这第一个模型:

```
. estimates store full
```

现在再估计一个简化模型(reduced model),只包括完整模型中自变量中的一部分(这种简化模型常被称为"嵌套"于完整模型)。最后,用命令 **lrtest *full*** 将嵌套模型对比以前所存的 *full* 模型来进行检验。比如(加上 **quietly** 前缀来取消输出,因为已经看到过这一输出了):

```
. quietly logistic any date
. lrtest full

likelihood-ratio test                          LR chi2(1)  =      3.28
(Assumption: . nested in full)                 Prob > chi2 =    0.0701
```

这个 **lrtest** 命令检验最近的(即指嵌套)模型对比以前由 **estimates store** 命令所存的模型。它将嵌套的两个最大似然估计模型进行总的检验:

$$\chi^2 = -2(\ln \mathbb{L}_1 - \ln \mathbb{L}_0) \qquad [10.6]$$

其中, $\ln \mathbb{L}_0$ 是第一个(含所有 x 变量的)模型的对数似然值,而 $\ln \mathbb{L}_1$ 为第二个(含 x 变量子集的)对数似然值。比较相应模型0和模型1得到的统计量,它服从卡方分布,自由度为这两个模型在复杂性上的差别(即排除的 x 变量数)。键入 **help lrtest** 可以得到关于这个命令的更多说明,它其实还可以用于任何Stata的最大似然估计程序(如

logit、**mlogit**、**stcox** 或许多其他程序)。这个总的卡方统计量在 **logit** 或 **logistic** 时例行输出(见公式[10.3]),是公式[10.6]的一个特例。

以前的 **lrtest** 例子完成了这个计算:

$$\chi^2 = -2[-12.991\,096 - (-11.350\,748)]$$
$$= 3.28$$

有 1 个自由度,相应的概率 $P=0.070\,1$,表明 *temp* 的影响在 $\alpha=0.10$ 水平统计性显著。由于样本规模很小以及第二类错误有致命后果,$\alpha=0.10$ 似乎是一个比通常的 $\alpha=0.05$ 的更谨慎的临界点。

条件效应标绘图

条件效应标绘图有助于理解 logistic 模型在概率方面意味着什么。这些标绘图背后的思路将以曲线表明,当维持所有其他的 *x* 变量在所选值上不变时(可选平均数、四分位数或极端值),模型预测的 *y* 变化如何表达为一个 *x* 变量的函数。比如,我们可以看到,当维持 *date* 在其第 25 个百分位不变时,任何热损事件的预测概率都是 *temp* 的函数。*date* 的第 25 个百分位可以由 **summarize date, detail** 求出,为 8 569,即 1983 年 6 月 18 日。

```
. quietly logit any date temp
. generate L1 = _b[_cons] + _b[date]*8569 + _b[temp]*temp
. generate Phat1 = 1/(1 + exp(-L1))
. label variable Phat1 "P(distress >= 1 | date = 8569)"
```

L1 就是预测的 logit 值;*Phat1* 为相应的 *distress*≥1 的预测概率,根据公式(10.5)计算。用类似的步骤可以将 *date* 固定于其第 75 个百分位上(9 341,即 1985 年 7 月 29 日)计算出任何 *distress* 的预测概率:

```
. generate L2 = _b[_cons] + _b[date]*9341 + _b[temp]*temp
. generate Phat2 = 1/(1 + exp(-L2))
. label variable Phat2 "P(distress >= 1 | date = 9341)"
```

我们现在可以画出对于两种 *date* 水平的 *temp* 与任何 *distress* 的概率之间的关系了,参见图 10.2。使用带许多纵向波段的中位数样条(**graph twoway mspline, bands(50)**)就产生出本图中的平滑曲线,作为近似的平滑 logistic 函数。

```
. graph twoway mspline Phat1 temp, bands(50)
      ||  mspline Phat2 temp, bands(50)
      ||  , ytitle("Probability of thermal distress")
      ylabel(0(.2)1, grid) xlabel(, grid)
         legend(label(1 "June 1983") label(2 "July 1985")
         rows(2) position(7) ring(0))
```

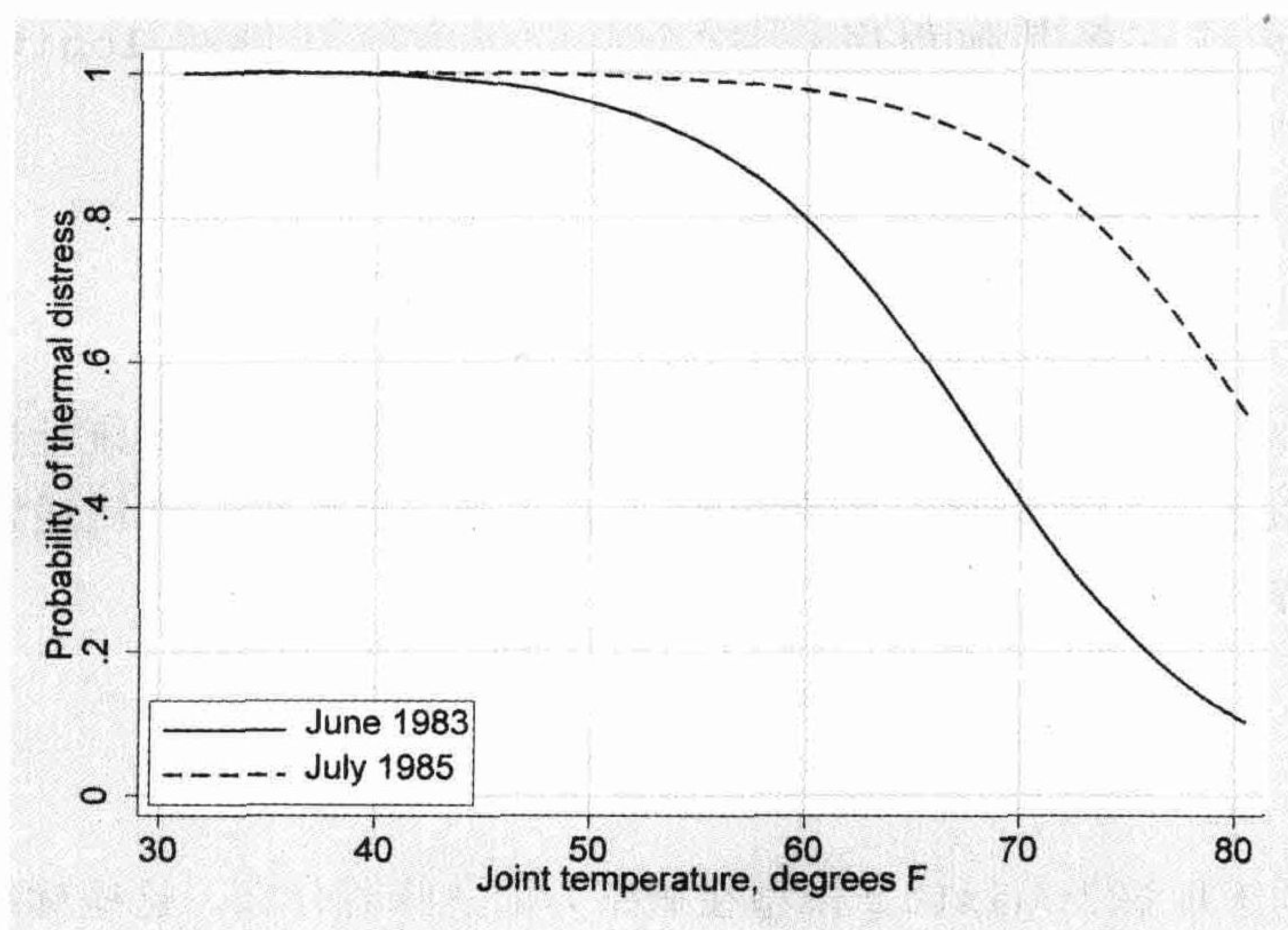


图 10.2

在较早的飞行(*date* = 8 569,左边的曲线)中,热损概率从约 80 °F 时的极低水平变为在 50 °F 时的接近为 1。然而,在较晚的航行(*date* = 9 341,右边的曲线)中,热损概率甚至在很暖和的天气时就超过了 0.5,在 70 °F 以下的飞行时热损概率便会接近于 1。注意,挑战者号的起飞温度为 31 °F,这将使它处于图 10.2 的左侧顶部。这个分析预测出助推结点几乎是肯定要损坏的。

诊断统计与标绘图

如前所述,用 **predict** 取得的 logistic 回归的影响及诊断统计并不是针对个别观测案例的,这与第 7 章中所讲的 OLS 回归诊断不同。相反,logistic 诊断是针对 *x* 模式而言的。然而,在航天飞机数据中,每一种 *x* 模式都是唯一的,即不存在两次飞行共享同样的 *date* 和 *temp*(很自然,也不会有两驾航天飞机在同一天起飞)。在使用 **predict** 之前,我们将安静地重新再拟合一下最近的模型,以保证这个模型就是我们所要的模型:

```
. quietly logistic any date temp

. predict Phat3
(option p assumed; Pr(any))

. label variable Phat3 "Predicted probability"

. predict dX2, dx2
(2 missing values generated)

. label variable dX2 "Change in Pearson chi-squared"

. predict dB, dbeta
(2 missing values generated)

. label variable dB "Influence"

. predict dD, ddeviance
(2 missing values generated)

. label variable dD "Change in deviance"
```

Hosmer 和 Lemeshow(2000)建议标绘图有助于理解这些诊断统计。要画出皮尔

森卡方变化对损坏概率(图 10.3)的图形,键入:

```
. graph twoway scatter dX2 Phat3
```

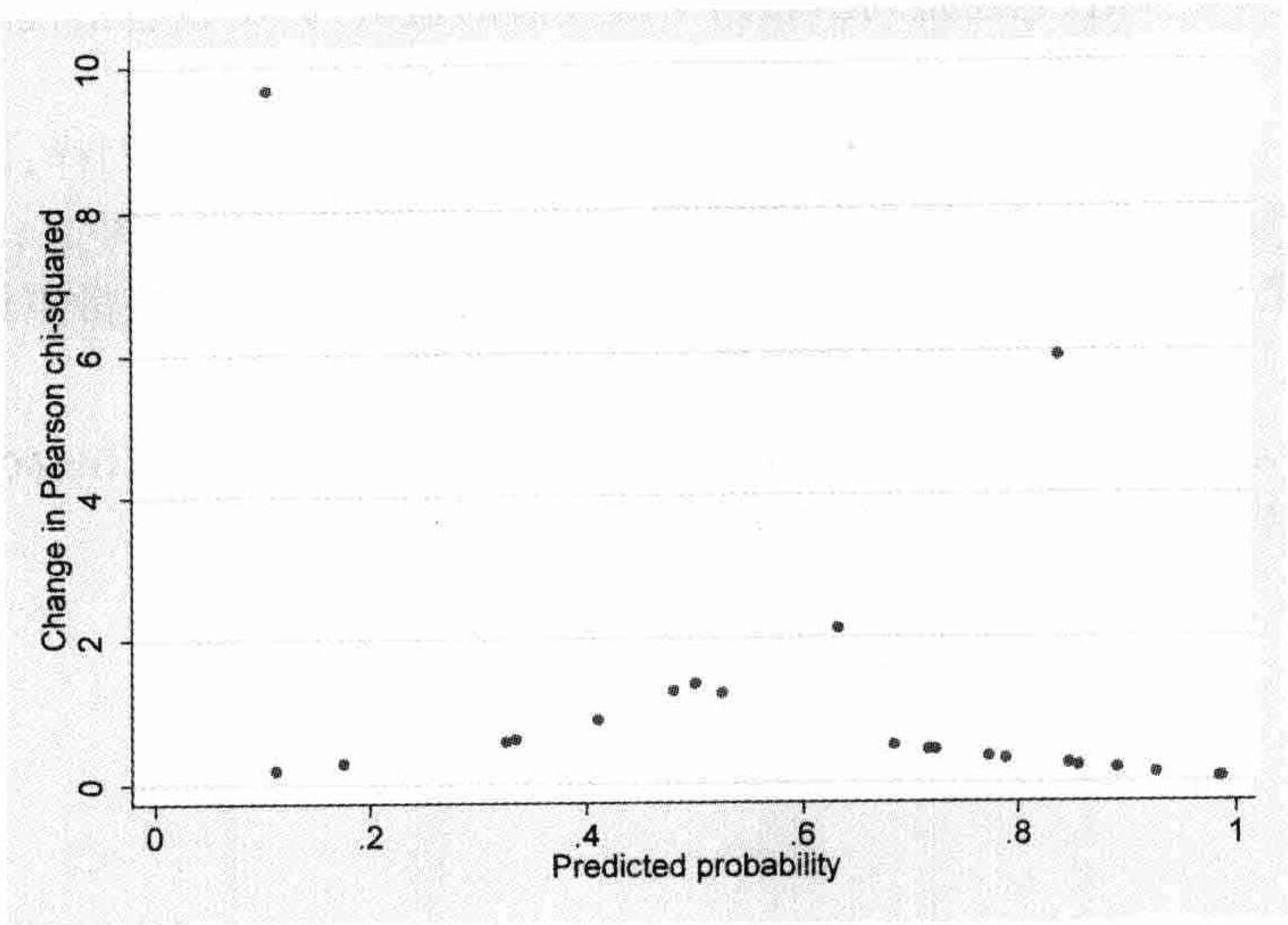


图 10.3

图中凸显出有两种拟合很差的 x 模式,在图的右上部和左上部。如果我们将标签包括于图中,就能直接从图中识别这两次航班(STS-2 和 STS 51-A),就像图 10.4 所示。

```
. graph twoway scatter dX2 Phat3, mlabel(flight) mlabsize(small)
```

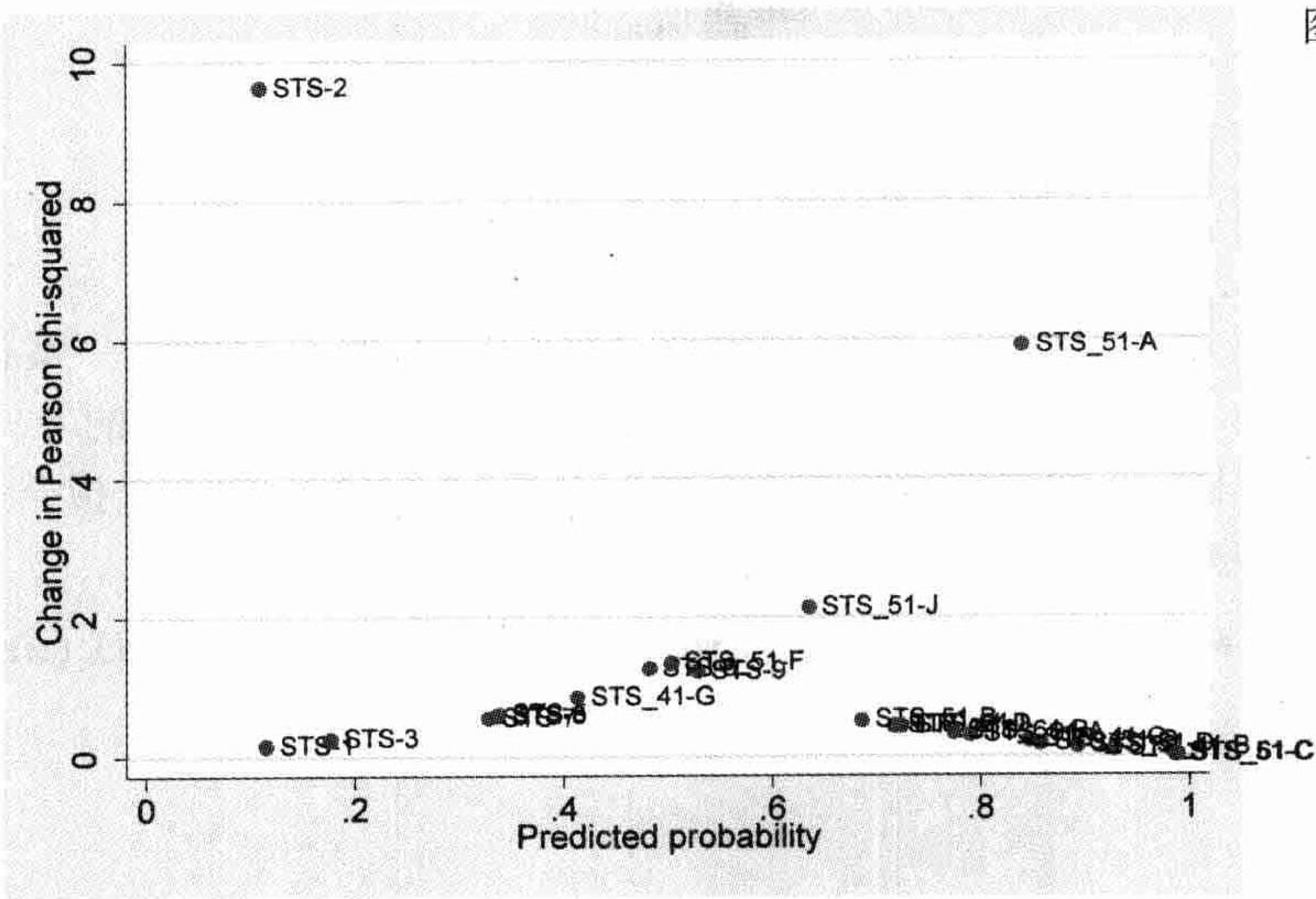


图 10.4

```
. list flight any date temp dX2 Phat3 if dX2 > 5
```

	flight	any	date	temp	dX2	Phat3
2.	STS-2	1	7986	70	9.630337	.1091805
4.	STS-4	.	8213	80	.	.0407113
14.	STS_51-A	0	9078	67	5.899742	.8400974
25.	STS_51-L	.	9524	31	.	.9999012

尽管起航很晚并且温度很冷,航班 STS 51-A 并没有发生热损(见图 10.2)。模型预测这次航班的损坏概率为 0.84。在图 10.4 中,所有属于右升曲线的观测点都有 *any* =

0，表示无热损发生。在左升曲线（有 *any* =1）的上头，尽管属于较早航班并且起航时天气较温和，航班 STS-2 还是发生了热损事件。模型预测其损坏概率只有 0.109。（由于 Stata 将缺失值作为“很大”数值，它还是列出有两个缺失值的航班，其中就包括挑战者号。它们就属于那些 *dX2* >5 的情况。）

类似的发现还可以从 *dD* 与预测概率的标绘图中取得，如图 10.5 所示。同样，因为航班 STS-2（左上端）和航班 STS 51-A（右上端）拟合很差，所以都很突出。图 10.5 示范了带标签散点图的一个变种。它没有像前面图 10.4 那样将航班号置于相应散点记号附近，而是不显示散点记号，并且将标签置于原先散点所在的位置上。

```
. graph twoway scatter dD Phat3, msymbol(i) mlabposition(0)
      mlabel(flight) mlabsize(small)
```

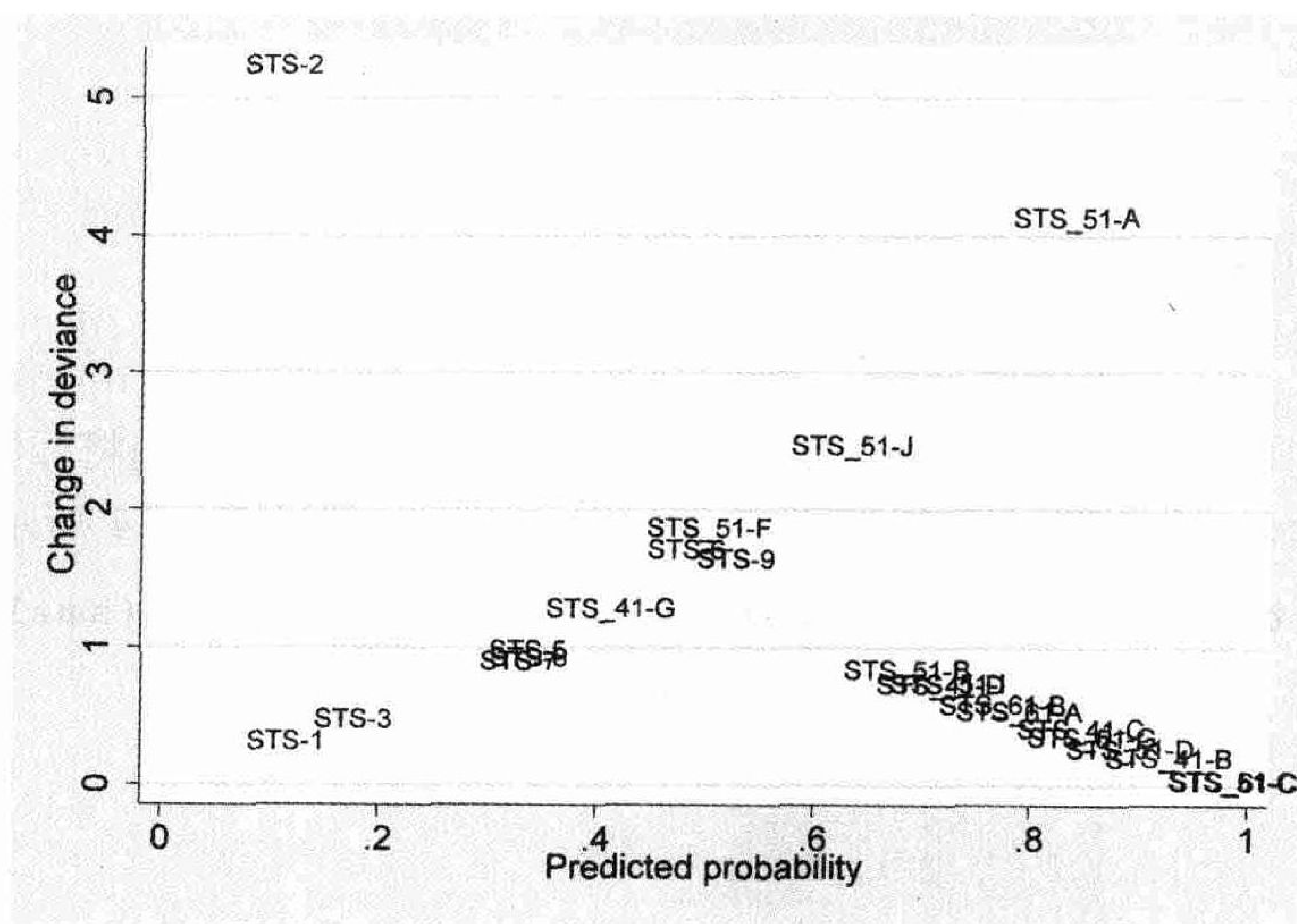


图 10.5

dB 测量某一 *x* 模式对 logistic 回归的影响，如同 Cook 的 *D* 测量某一案例对 OLS 估计影响一样。在 logistic 回归中，我们也能画出类似于图 7.7 那样的 OLS 诊断，将标绘记号的大小与其影响成比例，如图 10.6 所示。图 10.6 揭示出，两个拟合最差的观测同时就是最有影响的。

```
. graph twoway scatter dD Phat3 [aweight = dB], msymbol(oh)
```

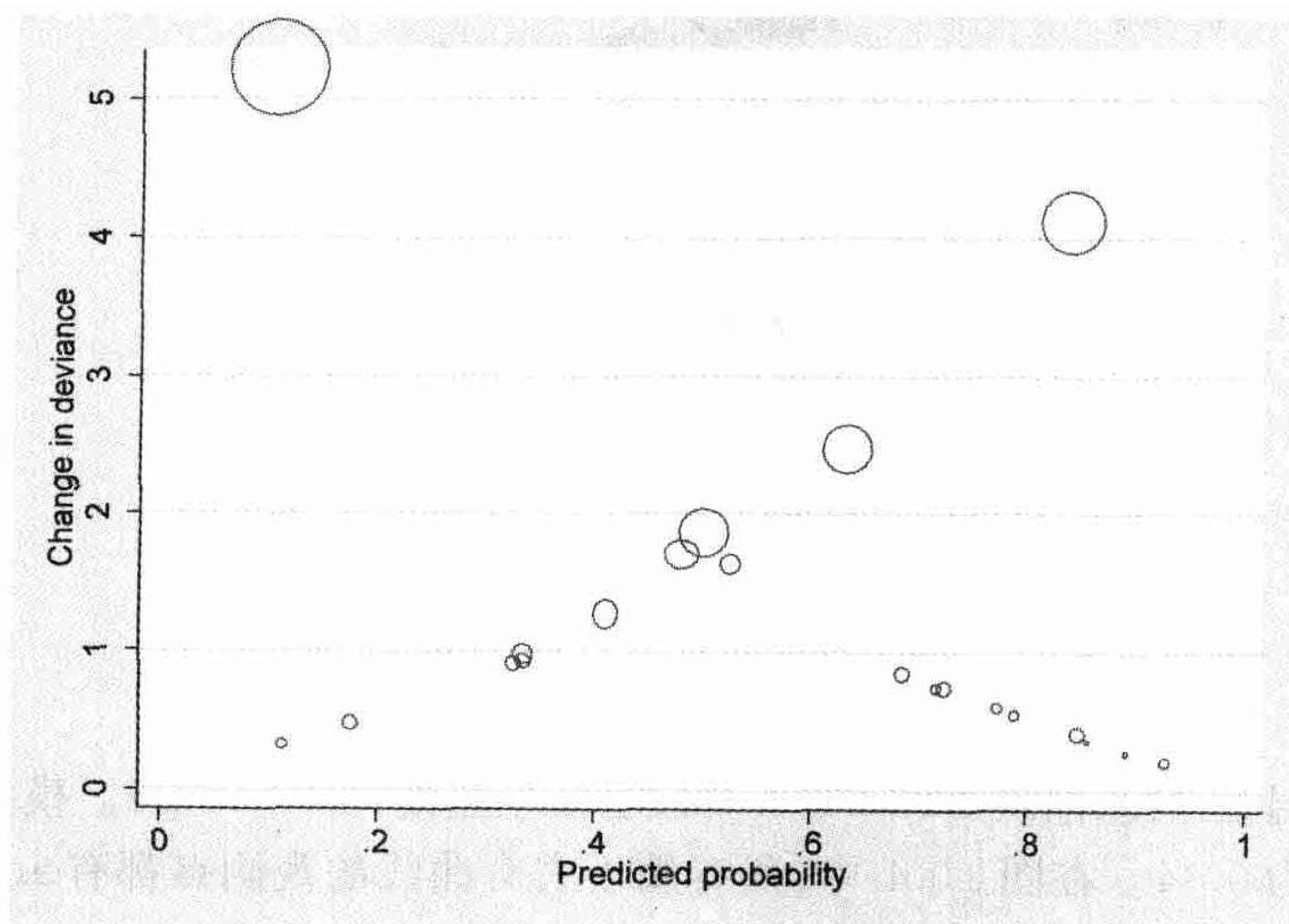


图 10.6

拟合差而又有影响的观测最值得特别关注,因为它们既与数据的主要模式矛盾、又将模型估计拉向与其相反的方向。当然,简单排除这些特异值可以取得对其他数据"更好的拟合",但是这便是一种循环论证。更深切的反应应当是研究这些特异值为什么不同寻常?为什么是航班 STS-2 而不是航班 STS 51-A 发生了助推结点损坏?寻求这种答案也许会导致研究人员发现以前忽视的变量或者按其他方式来定义模型。

对序次多分类 y 的 logistic 回归

logit 和 logistic 只能拟合含有两个类别{0,1}的 *y* 变量模型。因此,我们需要其他的方法来拟合那些 y 变量包含更多类别的情况。比如:

ologit 序次 logistic 回归(ordered logistic regression),其中 *y* 是序次变量。变量数值代表哪个类别并没有关系,但是较大数值要代表"较高"的类别。比如,*y* 的类别分为{1 ="差",2 = "中",3 = "好"}。

mlogit 多项 logistic 回归(multinomial logistic regression),其中 *y* 含有多个类别且类别并无顺序。比如,*y* 的类别分为{1 ="民主党",2 ="共和党",3 ="未申报"}。

如果 *y* 为{0,1},那么 **logit**(或 **logistic**)、**ologit** 和 **mlogit** 都会得到本质上一样的估计。

前面我们曾将三类的序次变量 *distress* 简化为二分变量 *any*,因为 **logit** 和 **logistic** 都要求{0,1}因变量。但是,**ologit** 就是设计出来分析像 *distress* 这样有多于两个类别的序次变量的。代表这些类别的具体数字编码并不重要,只要较大的数值代表着在测量上的"更多"即可。回顾一下,*distress* 的类别分为{0 ="无",1 = "1 或 2",2 ="3 及以上"}。

序次 logistic 回归表明,*date* 和 *temp* 都影响 *distress*,并且影响方向与我们前面分析的结果一样(即 *date* 有正影响,*temp* 为负影响)[11]:

```
. ologit distress date temp, nolog

Ordered logit estimates                           Number of obs   =         23
                                                  LR chi2(2)      =      12.32
                                                  Prob > chi2     =     0.0021
Log likelihood =  -18.79706                       Pseudo R2       =     0.2468

------------------------------------------------------------------------------
    distress |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        date |    .003286   .0012662     2.60   0.009      .0008043    .0057677
        temp |  -.1733752   .0834473    -2.08   0.038      -.336929   -.0098215
-------------+----------------------------------------------------------------
       _cut1 |   16.42813   9.554813          (Ancillary parameters)
       _cut2 |   18.12227   9.722293
------------------------------------------------------------------------------
```

似然比检验要比输出的渐近 *z* 检验更为准确。首先,我们用 **estimates store** 将刚刚估计的完整模型(有两个自变量)结果暂存于内存中,并任意指定这个模型称为 *A*。

```
. estimates store A
```

[11]【译注:以下输出表略有变化,底部多输出两个统计量。】

然后,再来拟合排除 *temp* 的简化模型,并将其结果暂存为模型 *B*,最后要求对简化模型(*B*)是否与完整模型(*A*)存在显著差异进行似然比检验:

```
. quietly ologit distress date
. estimates store B
. lrtest B A

likelihood-ratio test                                LR chi2(1)  =      6.12
(Assumption: B nested in A)                          Prob > chi2 =    0.0133
```

lrtest 输出中注明了它的假定,即模型 *B* 是嵌套于模型 *A* 的,也就是说模型 *B* 中的参数估计只是模型 *A* 中参数估计的一个子集,并且两个模型都是用同一套观测进行估计的(注意当有缺失时就可能不符了)。似然比检验表明,模型 *B* 的拟合显著较差。因为模型 *A* 只是多了 *temp* 这个自变量,所以似然比检验告诉我们,*temp* 的贡献是显著的。用类似的步骤也可以看到 *date* 也有显著影响。

```
. quietly ologit distress temp
. estimates store C
. lrtest C A

likelihood-ratio test                                LR chi2(1)  =     10.33
(Assumption: C nested in A)                          Prob > chi2 =    0.0013
```

estimates store 和 **lrtest** 命令提供比较嵌套的最大似然模型的灵活工具。键入 **help lrtest** 和 **help estimates** 来索取有关信息,以及更多高级的选项。

序次 logit 模型估计一个作为 *date* 和 *temp* 的线性函数的分值 *S*:

$$S = 0.003\,286\,date - 0.173\,375\,2\,temp$$

预测概率依赖于 *S* 的分值,再加上相对于估计的分割点的按 logistic 分布的扰动 *u*:

$P(distress = \text{“无”}) \quad = P(S + u \leq _cut1) \quad = P(S + u \leq 16.428\,13)$

$P(distress = \text{“1 或 2”}) \quad = P(_cut1 < S + u \leq _cut2) \quad = P(16.428\,13 < S + u \leq 18.122\,27)$

$P(distress = \text{“3 及以上”}) \quad = P(_cut2 < S + u) \quad = P(18.122\,27 < S + u)$

在执行 **ologit** 以后,**predict** 为因变量的每一个类别计算出预测概率。我们给 **predict** 提供了这些概率的命名。比如,用 *none* 来注明无损坏发生(*distress* 的第一个类别)的概率,用 *onetwo* 来注明 1 或 2 个损坏事件(*distress* 的第二个类别)的概率,用 *threeplus* 来注明 3 及以上事件(*distress* 的第三个、也是最后一个类别)的概率:

```
. quietly ologit distress date temp
. predict none onetwo threeplus
(option p assumed; predicted probabilities)
```

这就新建了三个变量:

```
. describe none onetwo threeplus

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
none            float  %9.0g                  Pr(distress==0)
onetwo          float  %9.0g                  Pr(distress==1)
threeplus       float  %9.0g                  Pr(distress==2)
```

为挑战者号最后一次航班(即这一数据的第 25 号)的预测概率令人不安:

```
. list flight none onetwo threeplus if flight == 25
```

```
     +-------------------------------------------+
     |   flight       none      onetwo   threep~s |
     |-------------------------------------------|
25.  | STS_51-L   .0000754   .0003346     .99959 |
     +-------------------------------------------+
```

基于对23个在挑战者号之前的航班的分析,我们的模型预测挑战者号几乎不可能无助推结点损坏事件($P = 0.000\ 075$),发生1或2处损坏的概率只稍微大一点($P = 0.000\ 3$),但是发生3处及以上损坏事件实际上是确定性的($P = 0.999\ 6$)。

参见Long(1997)或者Hosmer和Lemeshow(2000)对序次logistic回归的更多讨论以及有关技术。《基础参考手册》(*Base Reference Manual*)解释了Stata的有关执行工作。

多项logistic回归

当因变量的类别并没有自然顺序关系时,我们转向多项logistic回归(multinomial logistic regression),也称为多类(polytomous)logistic回归。**mlogit**命令也使其简单明了。当y仅有两个类别时,**mlogit**拟合的模型与**logistic**模型相同。然而,**mlogit**模型实际上要更为复杂。本节提供一个扩展的例子来解释**mlogit**结果,所用数据(*Nwarctic.dta*)来自于阿拉斯加西北极地自治区的高中学生调查(Hamilton和Seyfrit,1993)。

```
Contains data from C:\data\NWarctic.dta
  obs:          259                          NW Arctic high school students
                                               (Hamilton & Seyfrit 1993)
 vars:            3                          20 Jul 2005 10:40
 size:        2 590  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
life            byte   %8.0g       migrate    Expect to live most of life?
ties            float  %9.0g                  Social ties to community scale
kotz            byte   %8.0g       kotz       Live in Kotzebue or smaller
                                                village?
-------------------------------------------------------------------------------
```

变量*life*表明这些学生今后最愿意在什么地方长期生活,共区分3类:"same"即在本地区(西北极地);"other AK"即在阿拉斯加的其他地方;"leave AK"即在阿拉斯加以外:

```
. tabulate life, plot

  Expect to |
  live most |
   of life? |      Freq.
------------+------------+-----------------------------------------------------
       same |         92 |*****************************************
   other AK |        120 |*****************************************************
   leave AK |         47 |*********************
------------+------------+-----------------------------------------------------
      Total |        259
```

Kotzebue是西北极地地区的中心和最大城市(近3 000人口)。这些学生中三分之一以上居住于Kotzebue。其他人居住于只有200~700人的较小村落。Kotzebue学生较少恋家,并不想一辈子呆在本地,有较强倾向离开本州:

```
. tabulate life kotz, chi2

 Expect to |  Live in Kotzebue or
 live most |   smaller village?
  of life? |   village   Kotzebue |     Total
-----------+----------------------+----------
      same |        75         17 |        92
  other AK |        80         40 |       120
  leave AK |        11         36 |        47
-----------+----------------------+----------
     Total |       166         93 |       259

          Pearson chi2(2) =  46.2992   Pr = 0.000
```

mlogit 能够复制这个简单的分析(尽管其似然比卡方并不正好与 **tabulate** 得到的皮尔森卡方相等):

```
. mlogit life kotz, nolog base(1) rrr

Multinomial logistic regression                   Number of obs   =        259
                                                  LR chi2(2)      =      46.23
                                                  Prob > chi2     =     0.0000
Log likelihood = -244.64465                       Pseudo R2       =     0.0863

------------------------------------------------------------------------------
        life |        RRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
other AK     |
        kotz |   2.205882   .7304664     2.39   0.017     1.152687    4.221369
-------------+----------------------------------------------------------------
leave AK     |
        kotz |    14.4385   6.307555     6.11   0.000     6.132946    33.99188
------------------------------------------------------------------------------
(Outcome life==same is the comparison group)
```

base(1) 定义了 y 中的第 1 类(即 *life* = "same")作为比较的基准类别。选项 **rrr** 指示 **mlogit** 显示相对风险比,它类似于 **logistic** 给出的优势比。

回头再看 **tabulate** 输出,我们可以计算出,在 Kotzebue 学生中要"离开阿拉斯加"对比"留在原地"的发生比为:

$$P(\text{leave AK})/P(\text{same}) = (36/93)/(17/93) = 2.117\,647\,1$$

在其他学生中要"离开阿拉斯加"对比"留在原地"的发生比为:

$$P(\text{leave AK})/P(\text{same}) = (11/166)/(75/166) = 0.146\,666\,7$$

于是,Kotzebue 学生要"离开阿拉斯加"对比"留在原地"的发生比是其他学生相应发生比的 14.438 5 倍:

$$2.117\,647\,1/0.146\,666\,7 = 14.438\,5$$

这个倍数是两个发生比的比值,就是 **mlogit** 输出的相对风险比(14.438 5)。

总而言之,在其他条件相同时,y 的第 j 个类别在 x_k 条件下的相对风险比等于一个特定倍数,使 $y = j$(相对于 y = base)的预测发生比乘以这个特定倍数后,得到相应 $x_k + 1$ 条件下的发生比。换句话说,相对风险比 rrr_{jk} 就是当只有 x_k 变化而其他所有 x 不变时发生比变化的倍数:

$$rrr_{jk} \times \frac{P(y = j \mid x_k)}{P(y = \text{base} \mid x_k)} = \frac{P(y = j \mid x_k + 1)}{P(y = \text{base} \mid x_k + 1)}$$

变量 *ties* 是连续型测度,表示学生与家庭和社区之间的社会联系强度。我们将

ties 纳入模型作为第二个自变量：

```
. mlogit life kotz ties, nolog base(1) rrr

Multinomial logistic regression                   Number of obs   =        259
                                                  LR chi2(4)      =      91.96
                                                  Prob > chi2     =     0.0000
Log likelihood = -221.77969                       Pseudo R2       =     0.1717

------------------------------------------------------------------------------
        life |        RRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
other AK     |
        kotz |   2.214184   .7724996     2.28   0.023     1.117483    4.387193
        ties |   .4802486   .0799184    -4.41   0.000     .3465911    .6654492
-------------+----------------------------------------------------------------
leave AK     |
        kotz |   14.84604   7.146824     5.60   0.000     5.778907    38.13955
        ties |    .230262    .059085    -5.72   0.000     .1392531      .38075
------------------------------------------------------------------------------
(Outcome life==same is the comparison group)
```

这里的渐近 z 检验表明，描述两个 x 变量影响的 4 个相对风险比都显著地区别于 1.0。如果一个 y 变量有 J 个类别，那么 **mlogit** 建模时为每个自变量(x)的影响计算 $J-1$ 个相对风险比或系数，并因此进行 $J-1$ 个 z 检验，以评价对应每个自变量的两个或更多分别的虚无假设。似然比检验用于评价每个自变量的总影响。首先，我们将完整模型的结果加以暂存，并命名为 *full*：

```
. estimates store full
```

然后再排除一个 x 变量并拟合这一简化模型，并且进行相应似然比检验。比如，要检验自变量 *ties* 的影响，我们重复做一下排除 *ties* 后的回归：

```
. quietly mlogit life kotz

. estimates store no_ties

. lrtest no_ties full

likelihood-ratio test                                 LR chi2(2)  =     45.73
(Assumption: no_ties nested in full)                  Prob > chi2 =    0.0000
```

显然，*ties* 的影响是显著的。然后，我们再对变量 *kotz* 的影响进行类似的检验：

```
. quietly mlogit life ties

. estimates store no_kotz

. lrtest no_kotz full

likelihood-ratio test                                 LR chi2(2)  =     39.05
(Assumption: no_kotz nested in full)                  Prob > chi2 =    0.0000
```

如果我们的数据包含着缺失值，以上所示的三个 **mlogit** 命令就会分析三个有重叠的观测子集。完整模型就会只用在 *life*、*kotz*、*ties* 值上都不缺失的那些观测来回归；而只包含 *kotz* 的模型会将任何只有 *ties* 值缺失的观测回置于分析中；只包含 *ties* 的模型会将只有 *kotz* 值缺失的观测回置于分析中。当这些发生时，Stata 会返回一个错误提示说"observations differ."。在这种情况下，似然比检验实际上无效。这时，分析者必须在建模命令中加入 **if** 选择条件来筛选案例，比如：

```
. mlogit life kotz ties, nolog base(1) rrr
. estimates store full
. quietly mlogit life kotz if ties < .
. estimates store no_ties
. lrtest no_ties full
. quietly mlogit life ties if kotz < .
. estimates store no_kotz
. lrtest no_kotz full
```

或者干脆在分析之前就清除所有含缺失值的案例：

```
. drop if life >= . | kotz >= . | ties >= .
```

数据 *NWarctic.dta* 已经按此方式做过筛选，删除了那些含缺失值的观测案例。

两个自变量 *kotz* 和 *ties* 都能显著地预测 *life*。那么，我们还能对于这一输出说些什么呢？为了解释特定的影响，我们得知道 *life* ="same"（即留在本地）是基准类别。因此，相对风险比告诉我们：

- 学生中想迁往阿拉斯加别的地方对比想留在本地的发生比上，Kotzebue 的学生（即 *kotz* = 1）在调整了与社区联系强度影响后是其他学生的 2.21 倍（即提高了 121%）。
- 学生中想迁往阿拉斯加以外地方对比想留在本地的发生比上，Kotzebue 的学生（即 *kotz* = 1）在调整了与社区联系强度影响后是其他学生的 14.85 倍（即提高了 1 385%）。
- 学生中想迁往阿拉斯加别的地方对比想留在本地的发生比上，在控制了现居住地类型（Kotzebue 市或其他村落）的影响以后，每一个单位的社会联系强度增量（由于变量 *ties* 已经标准化了，其单位等于标准差）将使这一发生比变化 0.48 倍（即降低 52%）。
- 学生中想离开阿拉斯加对比想留在本地的发生比上，在控制了现居住地类型（Kotzebue 市或其他村落）的影响以后，每一个单位的社会联系强度增量将使这一发生比变化 0.23 倍（即降低 77%）。

predict 可以计算由 **mlogit** 模型取得的预测概率。选项 **outcome(#)** 指定我们想要 y 的哪一个类别的概率。比如，要得到 *life* ="leave AK"（离开阿拉斯加，即第 3 个类别）的预测概率：

```
. quietly mlogit life kotz ties
. predict PleaveAK, outcome(3)
(option p assumed; predicted probability)
. label variable PleaveAK "P(life = 3 | kotz, ties)"
```

按因变量的每种取值列出预测概率可以显示模型的拟合情况：

```
. table life, contents(mean PleaveAK) row
```

```
--------------------------
Expect to |
live most |
of life?  | mean(PleaveAK)
----------+---------------
     same |       .0811267
 other AK |       .1770225
 leave AK |       .3892264
          |
    Total |       .1814672
--------------------------
```

这些学生的少数(47/259=18%)期望离开阿拉斯加。即使对于那些实际已经选择期望离开这种答案的学生而言,这个模型计算其平均概率只有0.39。这反映出一个事实,即虽然我们的自变量存在很显著的影响,但在迁移计划方面仍有很大变异并没有得到解释。

条件效应标绘图有助于审视一个模型关于连续自变量的影响。我们可以应用估计的系数(而不是风险比)来计算概率,并画出其影响来:

```
. mlogit life kotz ties, nolog base(1)

Multinomial logistic regression                   Number of obs   =        259
                                                  LR chi2(4)      =      91.96
                                                  Prob > chi2     =     0.0000
Log likelihood = -221.77969                       Pseudo R2       =     0.1717

------------------------------------------------------------------------------
        life |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
other AK     |
        kotz |    .794884   .3488868     2.28   0.023     .1110784     1.47869
        ties |  -.7334513   .1664104    -4.41   0.000     -1.05961    -.407293
       _cons |    .206402   .1728053     1.19   0.232    -.1322902    .5450942
-------------+----------------------------------------------------------------
leave AK     |
        kotz |   2.697733   .4813959     5.60   0.000     1.754215    3.641252
        ties |  -1.468537   .2565991    -5.72   0.000    -1.971462   -.9656124
       _cons |  -2.115025   .3758163    -5.63   0.000    -2.851611   -1.378439
------------------------------------------------------------------------------
(Outcome life==same is the comparison group)
```

以下这些命令计算预测的 logit 值,然后再计算制作条件效应图时所需要的概率。*L2 villag* 代表对于居住于其他村落的学生的 *life* = 2(即要去阿拉斯加其他地方)的预测 logit 值;*L3 Kotz* 为代表对于居住于 Kotzebue 市的学生的 *life* = 3(即要离开阿拉斯加)的预测 logit 值;如此等等:

```
. generate L2villag = .206402 +.794884*0 -.7334513*ties
. generate L2kotz = .206402 +.794884*1 -.7334513*ties
. generate L3villag = -2.115025 +2.697733*0 -1.468537*ties
. generate L3kotz = -2.115025 +2.697733*1 -1.468537*ties
```

像其他 Stata 模型命令一样,**mlogit** 也将系数存为宏。比如,[2]_b[*kotz*]指模型第二个(*life* = 2)方程中 *kotz* 的系数。因此,我们也可以按以下方式来创建同样的预测 logit 值变量。*L2 v* 的值将与前面定义的 *L2 villag* 变量值相同,*L3 k* 的值与 *L3 kotz* 相同,如此等等:

```
. generate L2v = [2]_b[_cons] +[2]_b[kotz]*0 +[2]_b[ties]*ties
. generate L2k = [2]_b[_cons] +[2]_b[kotz]*1 +[2]_b[ties]*ties
. generate L3v = [3]_b[_cons] +[3]_b[kotz]*0 + [3]_b[ties]*ties
. generate L3k = [3]_b[_cons] +[3]_b[kotz]*1 + [3]_b[ties]*ties
```

然后,不论用哪一套 logit 值,我们再计算出相应的预测概率来:

```
. generate P1villag = 1/(1 +exp(L2villag) +exp(L3villag))
. label variable P1villag "same area"
. generate P2villag = exp(L2villag)/(1+exp(L2villag)+exp(L3villag))
. label variable P2villag "other Alaska"
. generate P3villag = exp(L3villag)/(1+exp(L2villag)+exp(L3villag))
. label variable P3villag "leave Alaska"
. generate P1kotz = 1/(1 +exp(L2kotz) +exp(L3kotz))
. label variable P1kotz "same area"
. generate P2kotz = exp(L2kotz)/(1 +exp(L2kotz) +exp(L3kotz))
. label variable P2kotz "other Alaska"
. generate P3kotz = exp(L3kotz)/(1 +exp(L2kotz) +exp(L3kotz))
. label variable P3kotz "leave Alaska"
```

图 10.7 和图 10.8 分别显示了其他村落和 Kotzebue 市学生的条件效应标绘图。

```
. graph twoway mspline P1villag ties, bands(50)
    || mspline P2villag ties, bands(50)
    || mspline P3villag ties, bands(50)
    || , xlabel(-3(1)3) ylabel(0(.2)1) yline(0 1) xline(0)
    legend(order(2 3 1) position(12) ring(0) label(1 "same area")
    label(2 "elsewhere Alaska") label(3 "leave Alaska") cols(1))
    ytitle("Probability")
```

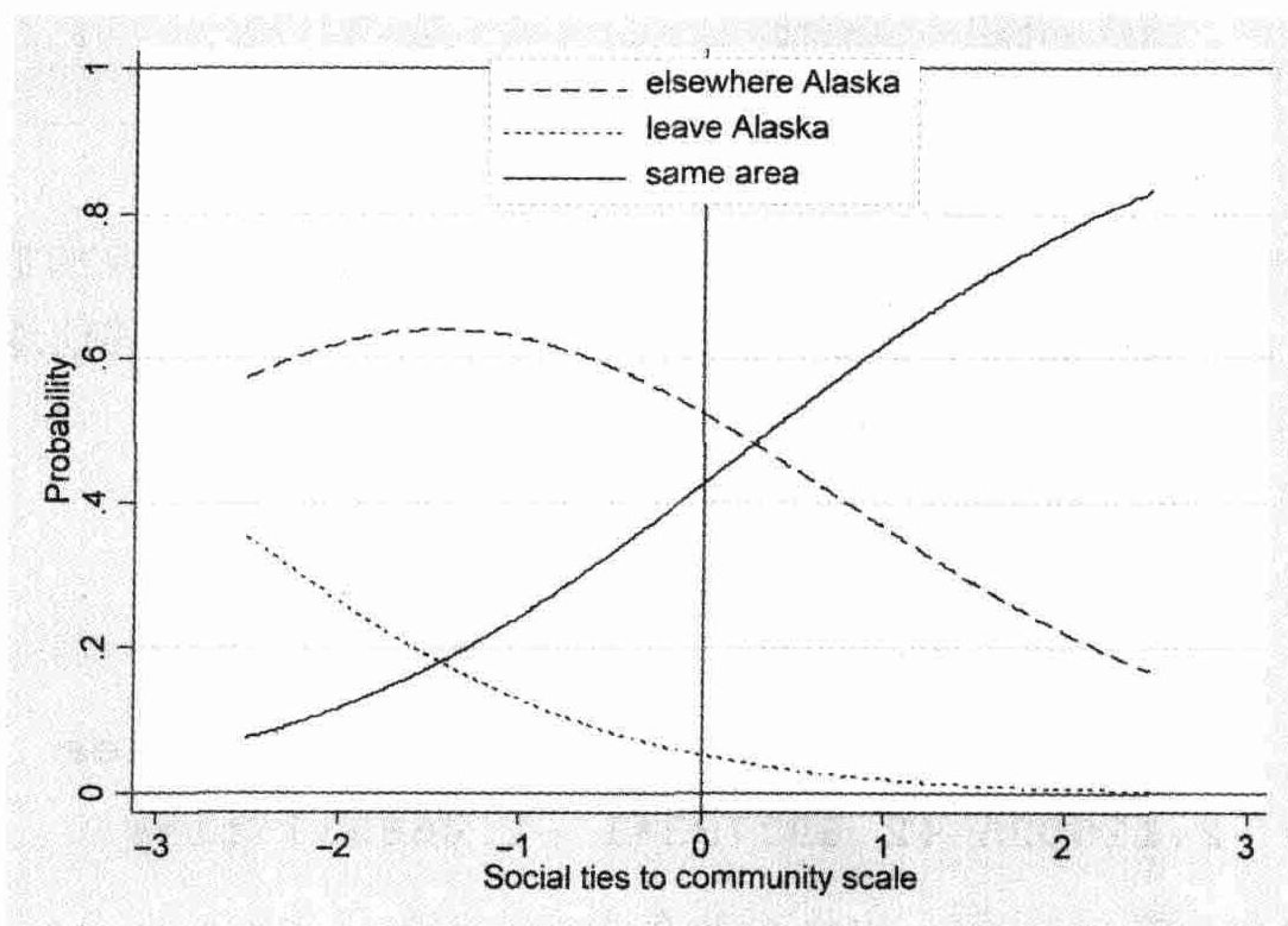


图 10.7

```
. graph twoway mspline P1kotz ties, bands(50)
    || mspline P2kotz ties, bands(50)
    || mspline P3kotz ties, bands(50)
    || , xlabel(-3(1)3) ylabel(0(.2)1) yline(0 1) xline(0)
    legend(order(3 2 1) position(12) ring(0) label(1 "same area")
    label(2 "elsewhere Alaska") label(3 "leave Alaska") cols(1))
    ytitle("Probability")
```

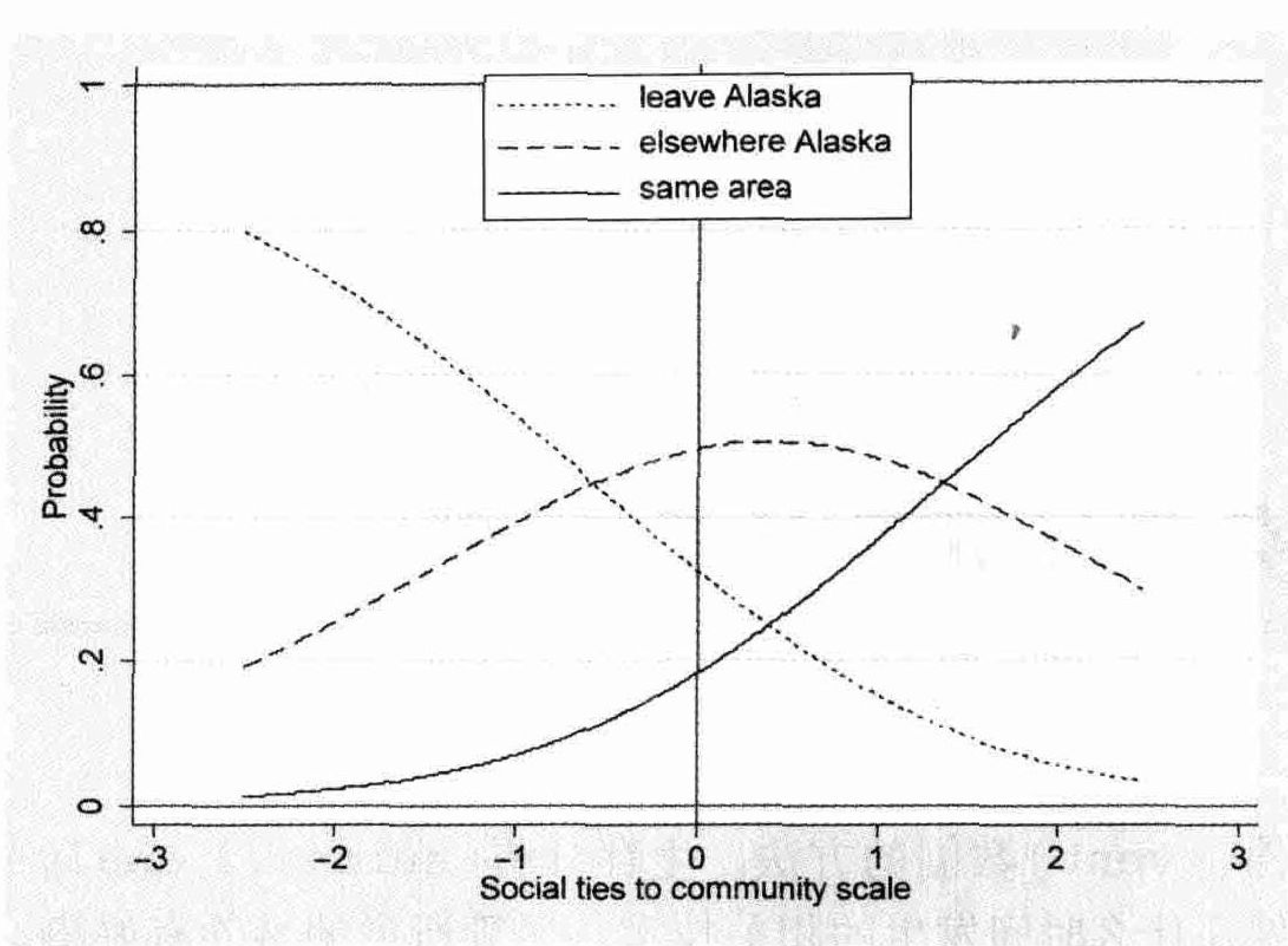


图 10.8

这些标绘图表明,在其他村落的学生中,社会联系(*ties*)能提高留在本地的概率,而不迁往阿拉斯加其他地方。这些村落学生中期望离开阿拉斯加的人相对较少。与此相反,在 Kotzebue 市的学生中,社会联系尤其对离开阿拉斯加有显著影响,而不是仅仅迁往本州的其他地方。只有当 Kotzebue 市的学生感到具有很强的社会联系时才会使他们倾向于选择留下来。

11 生存模型与事件计数模型

本章介绍的是分析事件(event)数据的方法。生存分析(survival analysis)包括好几种研究所关注事件在什么时间发生的相关技术。尽管所说的事件有好事、有坏事,但是按照习惯我们将事件称之为"失败"(failure)。于是,失败发生以前的时间就是"生存时间"(survival time)。生存分析在生物医学研究中十分重要,但是它也同样可以应用于像工程学到社会科学的其他领域。比如,可以建立一个关于失业者用多长时间才能找到工作的模型,或者关于人们在结婚前有多长时间未婚生活的模型。Stata 提供一系列的生存分析程序,本章只是示范了其中的几种。

我们还会简要地讨论泊松回归(Poisson regression)及其有关方法。这些方法并不关注生存时间,而是关注在特定时间段中事件的发生率或发生数。事件计数方法(event-count methods)包括泊松回归和负二项回归(negative binomial regression)。这些模型既可以用专门的命令来拟合,也可以通过一般化线性模型(generalized linear models, GLM)中提供的多种方法来拟合。

更多 Stata 的有关功能信息,请参阅《生存分析与流行病学梯度表参考手册》(*Survival Analsysis and Epidemiological Tables Reference Manual*)。键入 **help st** 也能得到在线总览。Selvin(1995)提供了关于生存分析和泊松回归很好的示范和介绍。本书也经过允许借用了好几个他的例子。其他一些对生存分析的很好的介绍还包括:Cleves、Gould 和 Gutierrez(2004)的基于 Stata 软件的一本书;Rosner(1995)书中的一章;Hosmer 和 Lemeshow(1999)以及 Lee(1992)的综合性著作。McCullagh 和 Nelder(1989)描述了一般化线性模型。Long(1997)的书中有一章是关于计数数据回归模型(包括了泊松回归和负二项回归),并且还包含了对一般化线性模型的一些讨论。在 Hardin 和 Hilbe(2001)的书里,对一般化线性模型进行广泛的讨论,并且论及当前的使用。

Stata 菜单中与本章最有关的部分包括:

Statistics-Survival analysis	生存分析
Graphics-Survival analysis graphs	生存分析作图
Statistics-Count outcomes	计数结果
Statistics-Generalized linear models (GLM)	一般化线性模型

关于流行病学梯度表的内容,并没有包括在本章中,有关信息可以通过 **help epitab** 获得,或者可以通过菜单取得咨询:

Statistics-Observational /Epi. analysis 观测及流行病学分析

命令示范

大多数 Stata 生存分析(**st** *)命令都要求数据已经提前用 **stset** 命令设置为生存时间(survival-time)(参见以下)。**stset** 只需要运行一次,并且所得数据随后要进行存盘。

```
. stset timevar, failure(failvar)
```

此命令用以识别单一记录(single-recorded)的生存时间数据。变量 *timevar* 提供了或是到某一特定事件(称为“失败”(failure))发生前、或是到观测结束时(称为“删截”(censoring))的消逝时间长度。变量 *failvar* 表明,相应 *timevar* 是以失败而结束(*failvar* =1)还是以删截而结束(*failvar* =0)。在这种数据中,一个观测对象只包含一条记录。在做任何进一步的 **st** * 命令之前,这一数据必须先要操作 **stset** 的处理。如果我们随后 **save**(存盘)这一数据,那么 **stset** 的定义将会同时被存盘。**stset** 创建的新变量 *_st*、*_d*、*_t* 以及 *_t0* 包含了有关编码信息,它们在随后执行的 **st** * 命令时是必须具备的。

```
. stset timevar, failure(failvar) id(patient) enter(time start)
```

这个命令识别多记录(multiple-recorded)的生存时间数据。在此例中,变量 *timevar* 提供到失败发生、或是到删截时的消逝时间。变量 *failvar* 则表明,它是以失败(1)还是以删截(0)而结束。*patient* 是一个识别码。在此数据中,同一个观测对象可以有多于一条的记录,但是相应记录必须具有同样的识别码。*start* 记录了每一对象进入观测的起始时间。

```
. stdes
```

这个命令描述生存时间数据,列出由 **stset** 设置的定义以及这一数据的其他特征。

```
. stsum
```

这个命令取得概要统计:历险时间总数,发生率,对象的数量,以及生存时间的百分位数。

```
. ctset time nfail ncensor nenter, by(ethnic sex)
```

这个命令识别计数时间数据(count-time data)。变量 *time* 是时间的测量;*nfail* 则是在 *time* 中失败的发生次数。我们还定义了 *ncensor*(即在 *time* 中删截观测的个数)和 *nenter*(即进入 *time* 中的观测个数),尽管这两个变量为可有可无的选项。在这些数据中,*ethnic* 和 *sex* 都是其他定义观测的特征分类变量。

```
. cttost
```

这个命令将前面用 **ctset** 命令设置的计数时间再转换成为生存时间数据形式,以便可以用 **st** * 命令来进行分析。

```
. sts graph
```

这个命令将提供 Kaplan-Meier 存活函数(Kaplan-Meier survivor function)的图形。为了可视化地比较两个或多个存活函数,比如,对分类变量 *sex* 不同性别作图,可在此命令中加上 **by(   )** 选项:

```
. sts graph, by(sex)
```

想要经过 Cox 回归对连续自变量如 *age*(年龄)的影响进行调整,可以使用 **adjustfor( )**选项:

```
. sts graph, by(sex) adjustfor(age)
```

注意:选项 **by( )**和 **adjustfor( )**在其他 **sts** 命令中也起类似的作用,比如,在 **sts list**、**sts generate** 以及 **sts test** 命令中。

```
. sts list
```

这个命令将列出 Kaplan-Meier 存活(或失败)函数。

```
. sts test sex
```

这个命令对 *sex* 的各类别的 Kaplan-Meier 存活函数进行等同性检验。

```
. sts generate survfunc = S
```

创建一个新的变量,任意指定变量名为 *survfunc*,用以存放估计的 Kaplan-Meier 存活函数。

```
. stcox x1 x2 x3
```

这个命令拟合一个 Cox 比例风险模型(Cox proportional hazard model),用失败之前的历险时间(time-to-failure)对连续或虚拟的自变量 *x1*、*x2*、*x3* 做回归。

```
. stcox x1 x2 x3, strata(x4) basechazard(hazard) robust
```

这个命令拟合一个 Cox 比例风险模型,用 *x4* 来做分层。将按组分别的基准累计风险函数(baseline cumulative hazard function)存为一个名为 *hazard* 的新变量。(基准存活函数估计可以用选项 **basesur**(***survive***)来取得,这个命令还能取得稳健(robust)标准误估计。参见第 9 章,或者参看《用户指南》中关于稳健标准误的更完全的解释。

```
. stphplot, by(sex)
```

这个命令根据前面刚估计的 **stcox** 模型为分类变量 *sex* 的每个类别画出 - ln(- ln(生存))相对于 ln(时间)的标绘图。要是取得了大致平行的曲线即表明支持了 Cox 模型关于风险比不随时间变化的假定。对 Cox 模型其他假定的检查可以使用命令 **stcoxkm**(比较 Cox 模型预测曲线和观测的 Kaplan-Meier 存活曲线)和 **stphtest**(根据 Schoenfeld 残差来进行检验)。有关命令和选项的说明,参见 **help stcox**。

```
. streg x1 x2, dist(weibull)
```

这个命令回归拟合 Weibull 分布模型(Weibull-distribution model),其中用失败前的历险时间对连续或虚拟的自变量 *x1* 和 *x2* 做回归。

```
. streg x1 x2 x3 x4, dist(exponential) robust
```

这个命令拟合指数分布(exponential distribution)模型,其中用失败前的历险时间对连续或虚拟的自变量 *x1* 至 *x4* 做回归。它能够取得对异方差性的稳健标准误估计。除了 Weibull 模型和指数模型外,**streg** 中关于其他 **dist( )**分布选项定义还包括对数正态分布(lognormal)、对数 logistic 分布(log-logistic)、Gompertz 分布或一般化 γ 分布(generalized gamma)。更多信息请参阅 **help streg**。

```
. stcurve, survival
```

在执行 **streg** 以后,此命令可画出相应模型在所有 *x* 变量的平均值上的生存函数。

```
. stcurve, cumhaz at(x3=50, x4=0)
```

在执行 **streg** 以后,此命令可画出相应模型在 *x1* 和 *x2* 的平均值上以及 *x3* 为 50 和 *x4* 为 0 时的累计风险函数。

```
. poisson count x1 x2 x3, irr exposure(x4)
```

此命令将事件计数变量 *count*(假定其服从泊松分布)对连续或虚拟的自变量 *x1*、*x2*、*x3* 做泊松回归。选项 **irr** 可使自变量的影响以发生率比(incidence rate ratio)的形式提供。如果所有观测的暴露期并不相同的话,用选项 **exposure( )** 指定一个表示暴露期数量的变量。

注意:泊松模型假定,不管一个事件发生了多少次,每一观测的事件概率都保持不变。如果事件概率并不保持不变,我们就应该考虑换用 **nbreg**(负二项回归,negative binomial regression)或 **gnbreg**(一般化负二项回归)。

```
. glm count x1 x2 x3,link(log)family(poisson)lnoffset(x4)eform
```

此命令完成与上述 **poisson** 例同样的回归,然而是作为一般化线性模型(GLM)中的一种模型。**glm** 可以拟合泊松模型、负二项模型、logit 模型以及许多其他类型的模型,取决于选项 **link( )** 指定的连接函数(link function)类型和选项 **family( )** 所指定使用的分布家族。

生存时间数据

生存时间数据包括:至少有一个变量测量每一观测对象在特定事件发生前所经历的时间。在有关文献中常常将所关注的事件统称为"失败",而不管其实际意义。当一个观测对象在数据收集结束时还未发生过事件时,就称这一观测为被"删截"了。命令 **stset** 为生存时间分析设置数据集,其任务是识别哪个变量用以测量时间,以及(如果必要)哪个变量表示观测是以失败或是删截为结束。这个数据还可以包括任意数量的其他测量变量或分类变量,一个观测对象(比如,医疗中的患者)可以有多于一个的观测记录。

为了示范 **stset** 的使用,我们以 Selvin(1995:453)关于 51 位诊断为 HIV 病毒携带者的研究作为例子。这一数据最开始是采用粗数据(raw-data)格式的文件 *aids.raw*,其数据看起来是这样的:

```
 1     1    1    34
 2    17    1    42
 3    37    0    47
   (第4至第50行被省略)
51    81    0    29
```

第 1 列值为案例号(1,2,3,…,51)。第 2 列说明在该患者从诊断以后至出现 AIDS 症状或至此研究观测结束已经过了多少个月(1,17,37,…)。当这位患者出现了 AIDS 症状(即失败),那么第 3 列就取 1 值,要是这位患者在研究结束时尚未出现症状(即删截),那么第 3 列就取 0 值。最后一列报告了该患者在诊断时的年龄。

我们可以用 **infile** 命令将粗数据读入内存,然后再为这些变量制作标签,并存为 Stata 格式的文件 *aids1.dta*:

```
. infile case time aids age using aids.raw, clear
(51 observations read)

. label variable case "Case ID number"

. label variable time "Months since HIV diagnosis"

. label variable aids "Developed AIDS symptoms"

. label variable age "Age in years"

. label data "AIDS (Selvin 1995:453)"

. compress
case was float now byte
time was float now byte
aids was float now byte
age was float now byte

. save aids1
file c:\data\aids1.dta saved
```

下一步是识别哪个变量测量时间,哪个变量表示失败或删截。尽管使用这种单一记录数据时并不必要,我们也可以注明哪个变量是每个案例的识别码(**id( )**)。在 **stset** 命令中,第一个提到的变量为测量时间的变量。然后,我们用 **failure( )** 来指定代表观测是失败(1)或是删截(0)的虚拟变量。执行 **stset** 以后,我们将数据做再次存盘以保留这些信息。

```
. stset time, failure(aids) id(case)

                id:  case
     failure event:  aids != 0 & aids < .
obs. time interval:  (time[_n-1], time]
 exit on or before:  failure

------------------------------------------------------------------------------
       51  total obs.
        0  exclusions
------------------------------------------------------------------------------
       51  obs. remaining, representing
       51  subjects
       25  failures in single failure-per-subject data
     3164  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =        97

. save, replace
file c:\data\aids1.dta saved
```

命令 **stdes** 将输出一个关于我们生存时间数据结构的简要描述。在这个简单例子中,每个对象只有一条记录,因此输出中的一些信息并不需要。

```
. stdes
         failure _d:  aids
   analysis time _t:  time
                 id:  case

                                   |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects               51
no. of records                51           1           1          1          1

(first) entry time                         0           0          0          0
(final) exit time                   62.03922           1         67         97

subjects with gap              0
time on gap if gap             0           .           .          .          .
time at risk                3164    62.03922           1         67         97

failures                      25    .4901961           0          0          1
------------------------------------------------------------------------------
```

命令 **stsum** 能够获取概要统计。在3 164 个人-月(person-month)中有25 个失败,于是得到发生率(incidence rate)为25/3 164 =0.007 901 4。由 Kaplan-Meier 存活函数(Kaplan-Meier survivor function,在后面讨论)可以推导出存活时间的百分位数。这一函数估计出,25% 的患者出现 AIDS 症状大约是诊断后 41 个月内,而 50% 的患者出现症状大约是在 81 个月内。在整个数据观测期间(共计 97 个月)出现 AIDS 症状的患者比例还不到 75% ,所以没有给出第 75 百分位数。

```
. stsum

         failure _d:  aids
   analysis time _t:  time
                 id:  case

         |               incidence       no. of    |------ Survival time -----|
         | time at risk     rate        subjects        25%       50%       75%
---------+---------------------------------------------------------------------
   total |         3164   .0079014            51         41        81         .
```

如果这个数据恰好包含一个分组或分类的变量,比如,性别变量 *sex*(0 为男,1 为女),我们就可以用以下命令形式分别为每一类别取得概要统计:

```
. stsum, by(sex)
```

以后各节将更为规范地描述比较两个或多个分组之间存活时间的方法。

计数时间数据

像 *aids1.dta* 这样的生存时间(**st**)数据包含着关于个人或事物的信息,用变量表示出每一个体的失败或删截发生的时间。还有一种数据称为计数时间(**ct**)数据,它所包含的是汇总数据,变量描述的是在时间 *t* 时发生失败或删截的个体的计数。比如,*diskdriv.dta* 中包含了关于25 只磁盘驱动器的假设检验信息。在1 200 小时检验完成时,除了5 只驱动器以外,其他全部驱动器都失败了。

```
Contains data from C:\data\diskdriv.dta
  obs:             6                          Count-time data on disk drives
 vars:             3                          21 Jul 2005 09:34
 size:            48 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
hours           int    %8.0g                  Hours of continuous operation
failures        byte   %8.0g                  Number of failures observed
censored        byte   %9.0g                  Number still working
-------------------------------------------------------------------------------
Sorted by:

. list

     +------------------------------+
     | hours   failures    censored |
     |------------------------------|
  1. |   200          2           0 |
  2. |   400          3           0 |
  3. |   600          4           0 |
  4. |   800          8           0 |
  5. |  1000          3           0 |
     |------------------------------|
  6. |  1200          0           5 |
     +------------------------------+
```

为了设置一个计数时间数据，我们需要按照顺序依次定义一个时间变量、一个失败计数变量以及一个删截计数变量。在执行 **ctset** 以后，用 **cttost** 命令自动地将我们的计数时间数据转换为生存时间格式。

```
. ctset hours failures censored

dataset name:  C:\data\diskdriv.dta
        time:  hours
     no. fail:  failures
     no. lost:  censored
    no. enter:  --                  (meaning all enter at time 0)

. cttost

(data are now st)

     failure event:  failures != 0 & failures < .
obs. time interval:  (0, hours]
 exit on or before:  failure
            weight:  [fweight=w]

------------------------------------------------------------------------------
        6  total obs.
        0  exclusions
------------------------------------------------------------------------------
        6  physical obs. remaining, equal to
       25  weighted obs., representing
       20  failures in single record/single failure data
    19400  total analysis time at risk, at risk from t =         0
                                 earliest observed entry t =         0
                                      last observed exit t =      1200

. list

     +-----------------------------------------------+
     | hours   failures   w   _st   _d     _t   _t0 |
     |-----------------------------------------------|
  1. |  1200          0   5     1    0   1200     0 |
  2. |   200          1   2     1    1    200     0 |
  3. |   400          1   3     1    1    400     0 |
  4. |   600          1   4     1    1    600     0 |
  5. |   800          1   8     1    1    800     0 |
     |-----------------------------------------------|
  6. |  1000          1   3     1    1   1000     0 |
     +-----------------------------------------------+

. stdes

         failure _d:  failures
   analysis time _t:  hours
             weight:  [fweight=w]

                                   |-------------- per subject --------------|
                        unweighted  unweighted              unweighted
Category                     total        mean        min       median        max
------------------------------------------------------------------------------
no. of subjects                  6
no. of records                   6           1          1           1          1

(first) entry time                           0          0           0          0
(final) exit time                          700        200         700       1200

subjects with gap                0
time on gap if gap               0
time at risk                  4200         700        200         700       1200

failures                         5    .8333333          0           1          1
------------------------------------------------------------------------------
```

命令 **cttost** 在其生存时间格式(**st**-format)的结果数据中定义了一套频数权数 *w*。**st** * 命令能够自动地识别这些权数并在任何生存时间分析中加以使用,因此这一数据现在被视为包括 25 个观测(25 个硬盘驱动器)而不再是以前的 6 个观测(即 6 个时期)。

```
. stsum

   failure time:  hours
 failure/censor:  failures
         weight:  [fweight=w]

       |               incidence       no. of     |---- Survival time ----|
       | time at risk     rate        subjects       25%        50%        75%
-------+---------------------------------------------------------------------
 total |        19400   .0010309            25       600        800       1000
```

Kaplan-Meier 存活函数

令 n_t 代表在时期 t 开始时的尚未失败且尚未删截的观测案例数。令 d_t 代表在时期 t 内这些观测中发生的失败数。Kaplan-Meier 存活函数关于生存超过时间 t 的估计是在时间 t 与在此以前各时期生存概率的连乘积:

$$S(t) = \prod_{j=t0}^{t} \{(n_j - d_j)/n_j\} \qquad [11.1]$$

比如,在上述的 AIDS 数据中,51 个患者中有一人在诊断之后只有一个月就产生了症状。这么早的时候还没有观测被删截,因此"存活"(其意义是不发展为 AIDS)超过 *time* = 1 时的概率为:

$$S(1) = (51 - 1)/51 = 0.980\,4$$

第二个患者在 *time* = 2 时产生症状,而第三个患者则是在 *time* = 9 时,于是有:

$$S(2) = 0.980\,4 \times (50 - 1)/50 = 0.960\,8$$

$$S(9) = 0.960\,8 \times (49 - 1)/49 = 0.941\,2$$

将 $S(t)$ 按时间 t 所绘的图就是 Kaplan-Meier 存活曲线,就像图 11.1 所示。在 Stata 中用 **sts graph** 命令便可自动绘出这样的图形。比如:

```
. use aids, clear
(AIDS (Selvin 1995:453))

. sts graph

          failure _d:  aids
    analysis time _t:  time
                  id:  case
```

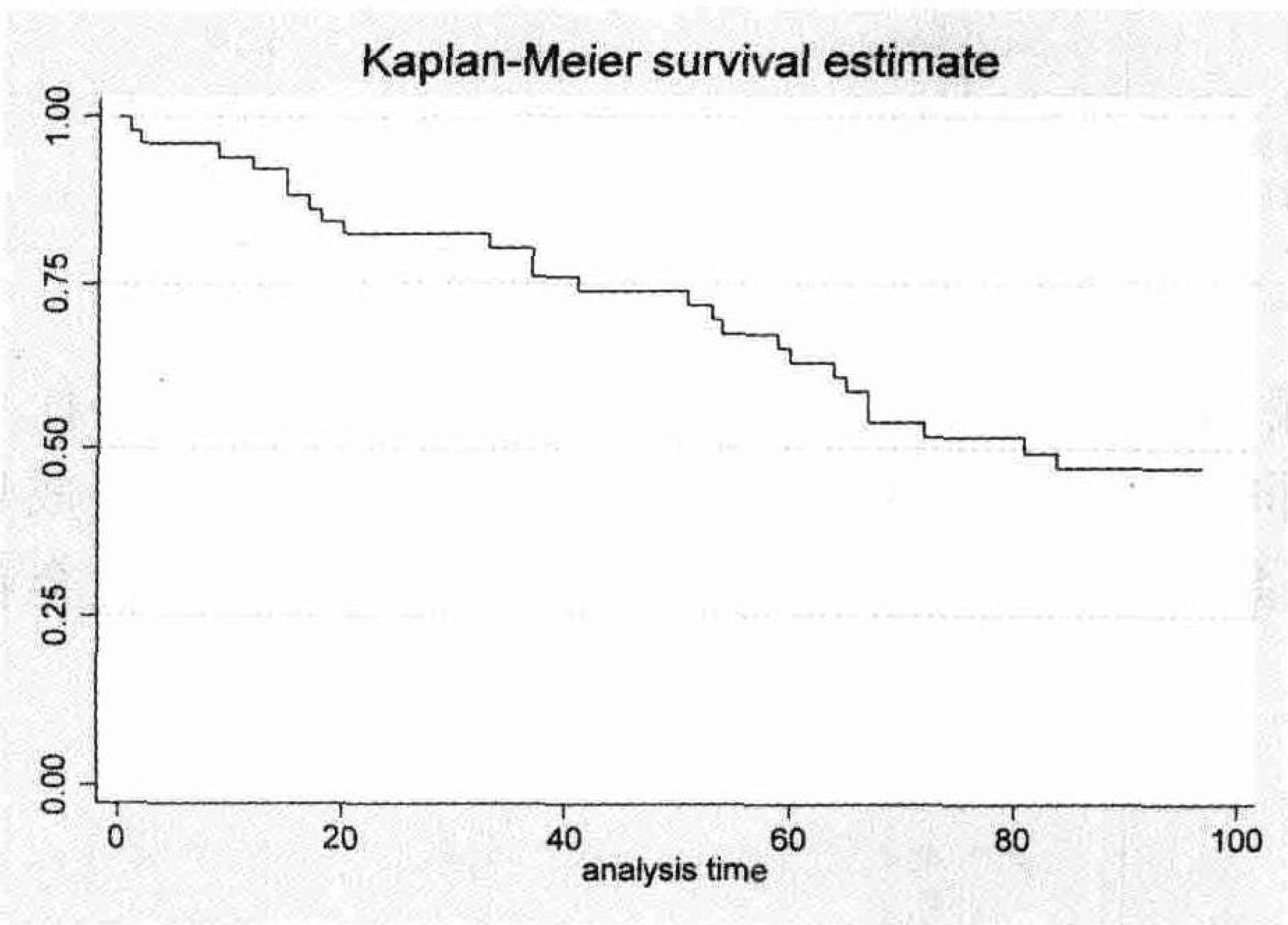


图 11.1

对于第二存活函数的例子,我们改用 *smoking1.dta* 中的数据,这一数据改编自 Rosner(1995)的著作。数据观测共有234名以前的吸烟者,都要尝试戒烟。大多数都没有成功。变量 *days* 记录了开始戒烟至恢复吸烟之间经历的天数。这个研究共持续了一年,用变量 *smoking* 表示每个人在研究结束前是否恢复吸烟了(*smoking* =1 代表"失败",而 *smoking* =0 代表"删截")。采用这套新数据,我们应当首先使用 **stset** 将这套数据转换为生存时间分析所需要的格式:

```
Contains data from C:\data\smoking1.dta
  obs:           234                          Smoking (Rosner 1995:607)
 vars:             8                          21 Jul 2005 09:35
 size:         3744 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              int    %9.0g                  Case ID number
days            int    %9.0g                  Days abstinent
smoking         byte   %9.0g                  Resumed smoking
age             byte   %9.0g                  Age in years
sex             byte   %9.0g       sex        Sex (female)
cigs            byte   %9.0g                  Cigarettes per day
co              int    %9.0g                  Carbon monoxide x 10
minutes         int    %9.0g                  Minutes elapsed since last cig
-------------------------------------------------------------------------------
Sorted by:

. stset days, failure(smoking)

     failure event:  smoking != 0 & smoking < .
obs. time interval:  (0, days]
 exit on or before:  failure

------------------------------------------------------------------------------
      234  total obs.
        0  exclusions
------------------------------------------------------------------------------
      234  obs. remaining, representing
      201  failures in single record/single failure data
    18946  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =       366
```

这一研究包括了110个男性和124个女性,两个性别的事件发生率显得很接近:

```
. stsum, by(sex)

         failure _d:  smoking
   analysis time _t:  days

         |               incidence       no. of    |------ Survival time -----|
sex      | time at risk     rate        subjects        25%       50%       75%
---------+---------------------------------------------------------------------
    Male |         8813   .0105526          110           4        15        68
  Female |        10133   .0106582          124           4        15        91
---------+---------------------------------------------------------------------
   total |        18946   .0106091          234           4        15        73
```

图11.2确认了这种相似性,表明男性和女性之间在存活函数上几乎没有什么差别。这就是说,两种性别在大约以同样的速率恢复吸烟。戒烟者的存活概率在戒烟后的前30天中下降非常迅速。不管是哪一性别,能够戒烟超过一整年的存活机会都不到15%。

```
. sts graph, by(sex)

        failure _d:  smoking
  analysis time _t:  days
```

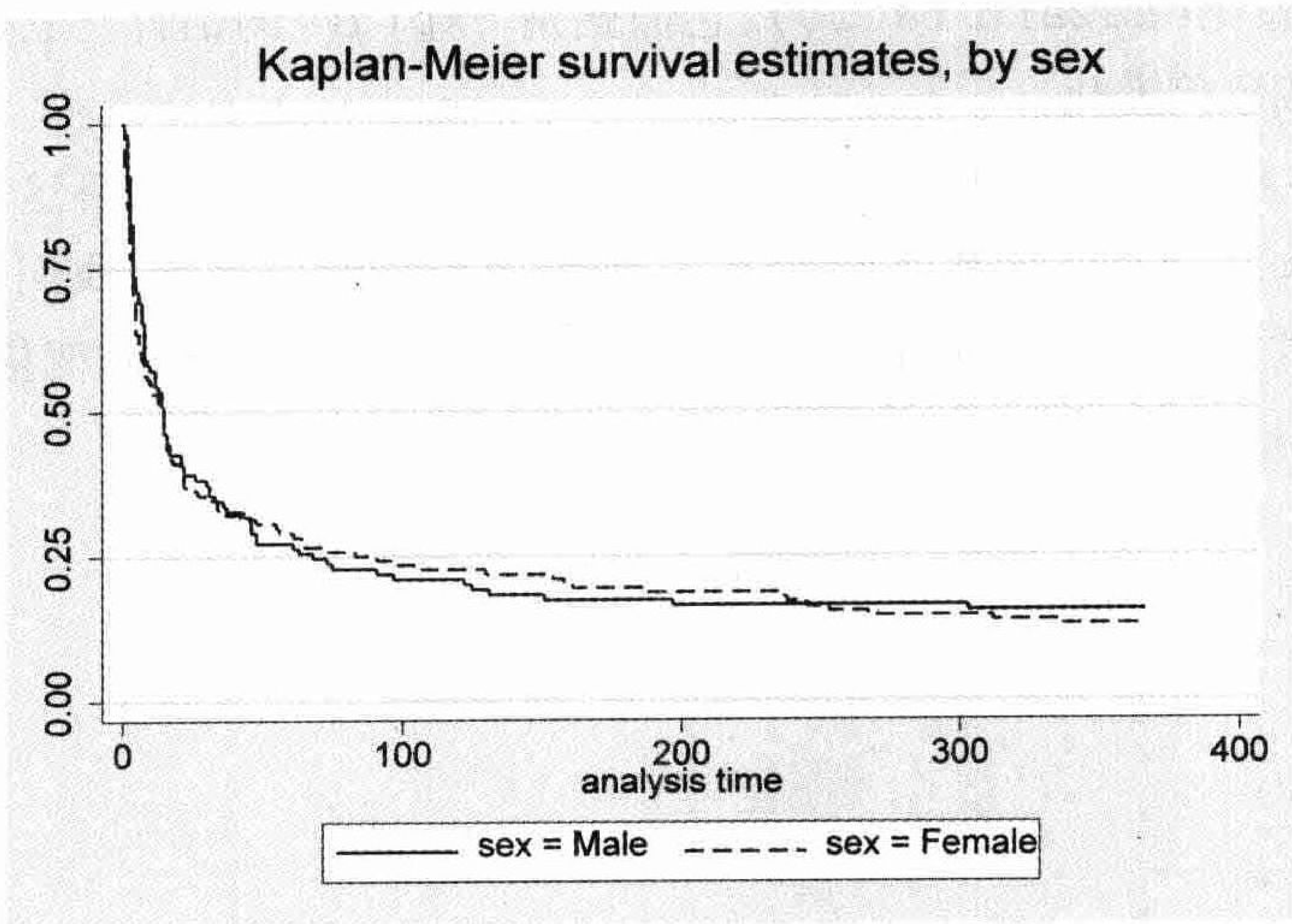


图 11.2

我们还可以使用对数秩检验(log-rank test)规范地检验存活函数的等同性。毫不奇怪,这一检验并没有发现男性与女性之间在重犯吸烟上有显著差别($P = 0.677\ 2$)。

```
. sts test sex

        failure _d:  smoking
  analysis time _t:  days


Log-rank test for equality of survivor functions

       |   Events         Events
sex    |  observed       expected
-------+-------------------------
Male   |        93          95.88
Female |       108         105.12
-------+-------------------------
Total  |       201         201.00

             chi2(1) =       0.17
             Pr>chi2 =     0.6772
```

Cox 比例风险模型

回归方法可以让我们的生存分析更进一步地检查多元的连续或分类自变量的影响。一个广为应用的方法就是应用比例风险模型(proportional hazard model)的 Cox 回归。在时间 t 上的失败风险率(hazard rate for failure)定义为:

$$h(t) = \frac{\text{在时间 } t \text{ 和 } t+\Delta t \text{ 之间的失败概率}}{(\Delta t)(\text{时间 } t \text{ 以后的失败概率})} \qquad [11.2]$$

我们将这个风险率建模为在时间 t 上的基准风险(baseline hazard,标为 h_0)和一个或多个 x 变量影响的函数

$$h(t) = h_0(t)\exp(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k) \qquad [11.3a]$$

或者,等价地表达为:

$$\ln[h(t)] = \ln[h_0(t)] + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \qquad [11.3b]$$

所谓“基准风险”指当一个观测的所有 x 变量都等于 0 时的风险。Cox 回归以非参数方式对这个风险作出估计,并且取得公式[11.3]中那些 β 参数的最大似然估计。Stata 的 **stcox** 程序通常是报告风险比(hazard ratio),它们就是 exp(β)的估计。它们表示相对基准风险率而言的成比例变化。

年龄会不会影响 AIDS 症状的发生呢? 数据 *aids.dta* 包括了有助于回答这个问题的信息。请注意,**stcox** 命令与大多数其他 Stata 的模型拟合命令不同,我们只需要在这个命令中列出自变量即可。生存分析的因变量,包括时间变量以及删截变量,都是从 **stset** 数据中自动提取的。

```
. use aids
(AIDS (Selvin 1995:453))

. stcox age, nolog

         failure _d:  aids
   analysis time _t:  time
                 id:  case

Cox regression -- Breslow method for ties

No. of subjects =          51                  Number of obs    =        51
No. of failures =          25
Time at risk    =        3164
                                               LR chi2(1)       =      5.00
Log likelihood  =  -86.576295                  Prob > chi2      =    0.0254

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.084557   .0378623     2.33   0.020      1.01283    1.161363
------------------------------------------------------------------------------
```

我们可以通过参照年龄分别为 a 岁和 $a+1$ 岁的两名 HIV 阳性的人来解释估计出的风险比 1.084 557。它表明,年龄大 1 岁,那么在近期发生 AIDS 症状的风险就会提高 8.5%(即他们两人各自风险的比值为 1.084 557)。这个比值显著地($P=0.020$)区别于 1。如果我们想要按 5 岁年龄之差来陈述我们的发现,便可以将这一风险比做 5 次方:

```
. display exp(_b[age])^5
1.5005865
```

于是,如果第二个人比第一个人年龄大 5 岁的话,那么其 AIDS 发生的风险会比前者高出 50%。我们还可以换一种方式来取得同样的结果(并且还可以取得新的置信区间),就是先建立一个新版本的年龄变量,它以 5 岁为测量单位,然后再重新做一次回归。下列命令中的 **nolog noshow** 选项取消了迭代过程日志和生存数据描述的显示。

```
. generate age5 = age/5
. label variable age5 "age in 5-year units"
. stcox age5, nolog noshow
```

```
Cox regression -- Breslow method for ties

No. of subjects =           51                  Number of obs    =        51
No. of failures =           25
Time at risk    =         3164
                                                LR chi2(1)       =      5.00
Log likelihood  =   -86.576295                  Prob > chi2      =    0.0254

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        age5 |   1.500587   .2619305     2.33   0.020     1.065815    2.112711
------------------------------------------------------------------------------
```

与常规回归类似,Cox 模型也能包括多个自变量。数据 *heart.dta* 中包含了 Selvin(1995)研究 35 位胆固醇水平极高的患者的生存时间数据。变量 *time* 提供了每一位患者的观察天数。*coronary* 则代表在这一时间冠心病是否发作(*coronary* = 1 为是,*coronary* = 0 为否)。这一数据中还包括了胆固醇水平和其他一些心脏病的影响因素。数据 *heart.dta* 以前已经由 **stset time, failure(*coronary*)** 命令设置为生存时间分析格式了,所以我们可以直接开始进行生存分析。

```
. describe patient - ab

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
patient         byte   %9.0g                  Patient ID number
time            int    %9.0g                  Time in days
coronary        byte   %9.0g                  Coronary event (1) or none (0)
weight          int    %9.0g                  Weight in pounds
sbp             int    %9.0g                  Systolic blood pressure
chol            int    %9.0g                  Cholesterol level
cigs            byte   %9.0g                  Cigarettes smoked per day
ab              byte   %9.0g                  Type A (1) or B (0) personality

. stdes

         failure _d:  coronary
   analysis time _t:  time

                                   |-------------- per subject --------------|
Category                   total        mean         min     median        max
------------------------------------------------------------------------------
no. of subjects               35
no. of records                35           1           1          1          1

(first) entry time                         0           0          0          0
(final) exit time                   2580.629         773       2875       3141

subjects with gap              0
time on gap if gap             0
time at risk               90322    2580.629         773       2875       3141

failures                       8    .2285714           0          0          1
------------------------------------------------------------------------------
```

Cox 回归发现,胆固醇水平(*chol*)和吸烟(*cigs*)都显著地提高了冠心病发作的风险。与直觉相违背,体重(*weight*)却显得可以降低这种风险。血压(*sbp*)和 A/B 两种个性(*ab*)都没有显著的净效应。

```
. stcox weight sbp chol cigs ab, noshow nolog
```

```
Cox regression -- no ties

No. of subjects =           35                  Number of obs   =         35
No. of failures =            8
Time at risk    =        90322
                                                LR chi2(5)      =      13.97
Log likelihood  =   -17.263231                  Prob > chi2     =     0.0158

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |   .9349336   .0305184    -2.06   0.039     .8769919    .9967034
         sbp |   1.012947   .0338061     0.39   0.700     .9488087    1.081421
        chol |   1.032142   .0139984     2.33   0.020     1.005067    1.059947
        cigs |   1.203335   .1071031     2.08   0.038     1.010707    1.432676
          ab |    3.04969   2.985616     1.14   0.255     .4476492    20.77655
------------------------------------------------------------------------------
```

进行模型估计之后，**stcox** 还能够建立新变量来保留估计的基准累计风险(baseline cumulative hazard)和存活函数(survivor function)。因为“基准”是指所有 *x* 变量等于 0 的情况，所以我们应当对所有变量重新对中(re-center)以便使它们的 0 值具有实际意义。一个患者如果要是体重为 0 磅、或血压为 0，都不能提供一个有用的比较基准。参照我们数据中的实际最小值，我们可以对体重(*weight*)这个变量加以改变，使其 0 值实际代表 120 磅，并且将血压(*sbp*)的 0 值代表 100，将胆固醇水平(*chol*)的 0 值代表 340：

```
. summarize patient - ab

    Variable |     Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------
     patient |      35          18    10.24695          1         35
        time |      35    2580.629    616.0796        773       3141
    coronary |      35    .2285714     .426043          0          1
      weight |      35    170.0857    23.55516        120        225
         sbp |      35    129.7143    14.28403        104        154
-------------+-----------------------------------------------------
        chol |      35    369.2857    51.32284        343        645
        cigs |      35    17.14286    13.07702          0         40
          ab |      35    .5142857    .5070926          0          1

. replace weight = weight - 120
(35 real changes made)

. replace sbp = sbp - 100
(35 real changes made)

. replace chol = chol - 340
(35 real changes made)

. summarize patient - ab

    Variable |     Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------
     patient |      35          18    10.24695          1         35
        time |      35    2580.629    616.0796        773       3141
    coronary |      35    .2285714     .426043          0          1
      weight |      35    50.08571    23.55516          0        105
         sbp |      35    29.71429    14.28403          4         54
-------------+-----------------------------------------------------
        chol |      35    29.28571    51.32284          3        305
        cigs |      35    17.14286    13.07702          0         40
          ab |      35    .5142857    .5070926          0          1
```

现在，所有 *x* 变量的 0 值就都有具有实际意义了。要建立新变量来分别保留基准存活

估计和累计风险函数估计,我们就再做一次上述回归,但是在命令中加上 **basesurv( )**和 **basechaz( )**选项:

```
. stcox weight sbp chol cigs ab, noshow nolog basesurv(survivor)
      basechaz(hazard)

Cox regression -- no ties

No. of subjects =           35                 Number of obs    =        35
No. of failures =            8
Time at risk    =        90322
                                               LR chi2(5)       =     13.97
Log likelihood  =   -17.263231                 Prob > chi2      =    0.0158

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |   .9349336   .0305184    -2.06   0.039     .8769919    .9967034
         sbp |   1.012947   .0338061     0.39   0.700     .9488087    1.081421
        chol |   1.032142   .0139984     2.33   0.020     1.005067    1.059947
        cigs |   1.203335   .1071031     2.08   0.038     1.010707    1.432676
          ab |    3.04969   2.985616     1.14   0.255     .4476492    20.77655
------------------------------------------------------------------------------
```

注意,那 3 个重新对中的 x 变量并没有对风险比、标准误以及其他估计产生任何影响。这一命令建立了两个新变量,我们随意地将其命名为 *survivor* 和 *hazard*。要想给基准存活函数作图,我们以 *time* 为横轴画出 *survivor* 的标绘图,并且将数据点之间做阶梯状连线,如图 11.3 所示。

```
. graph twoway line survivor time, connect(stairstep) sort
```

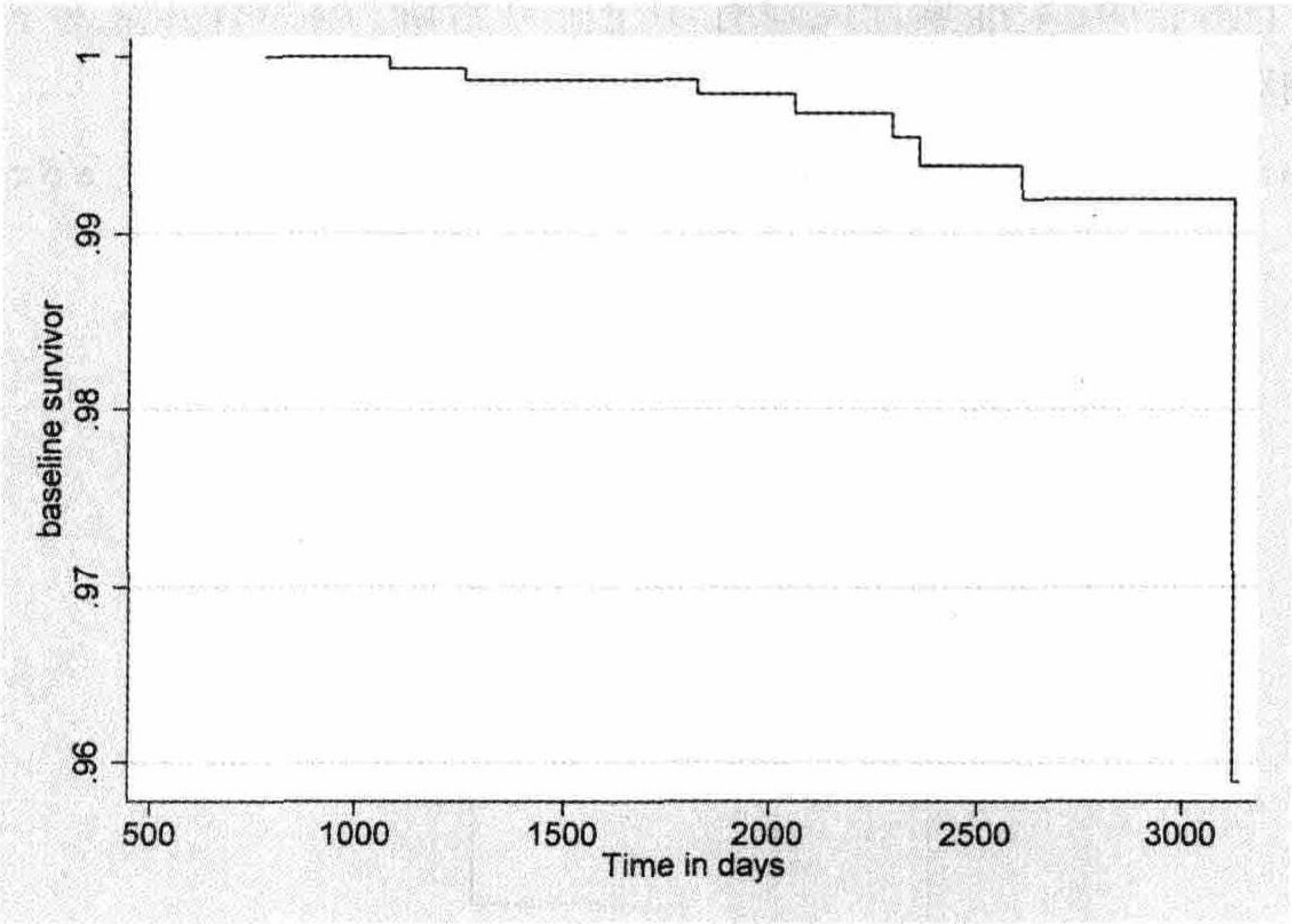


图 11.3

基准存活函数揭示了具有“0”体重(120 磅)、“0”血压(100)、每天吸“0”枝香烟的 B 类个性的患者的存活概率是随时间而降低的。尽管这一概率在最右侧表面看起来极迅速地下降,但是要注意,这一概率实际上只是从 1 下降到大约 0.96 而已。如果考虑自变量方面的不良影响,那么存活概率将下降得更快。

同一基准存活函数图也可以不用 **stcox** 命令而以另外的方式取得。替换方法取得的图 11.4 使用了 **sts graph** 命令加上 **adjustfor( )**的调整选项,并列上要调整的自变量名称:

```
. sts graph, adjustfor(weight sbp chol cigs ab)
```

```
      failure _d:  coronary
analysis time _t:  time
```

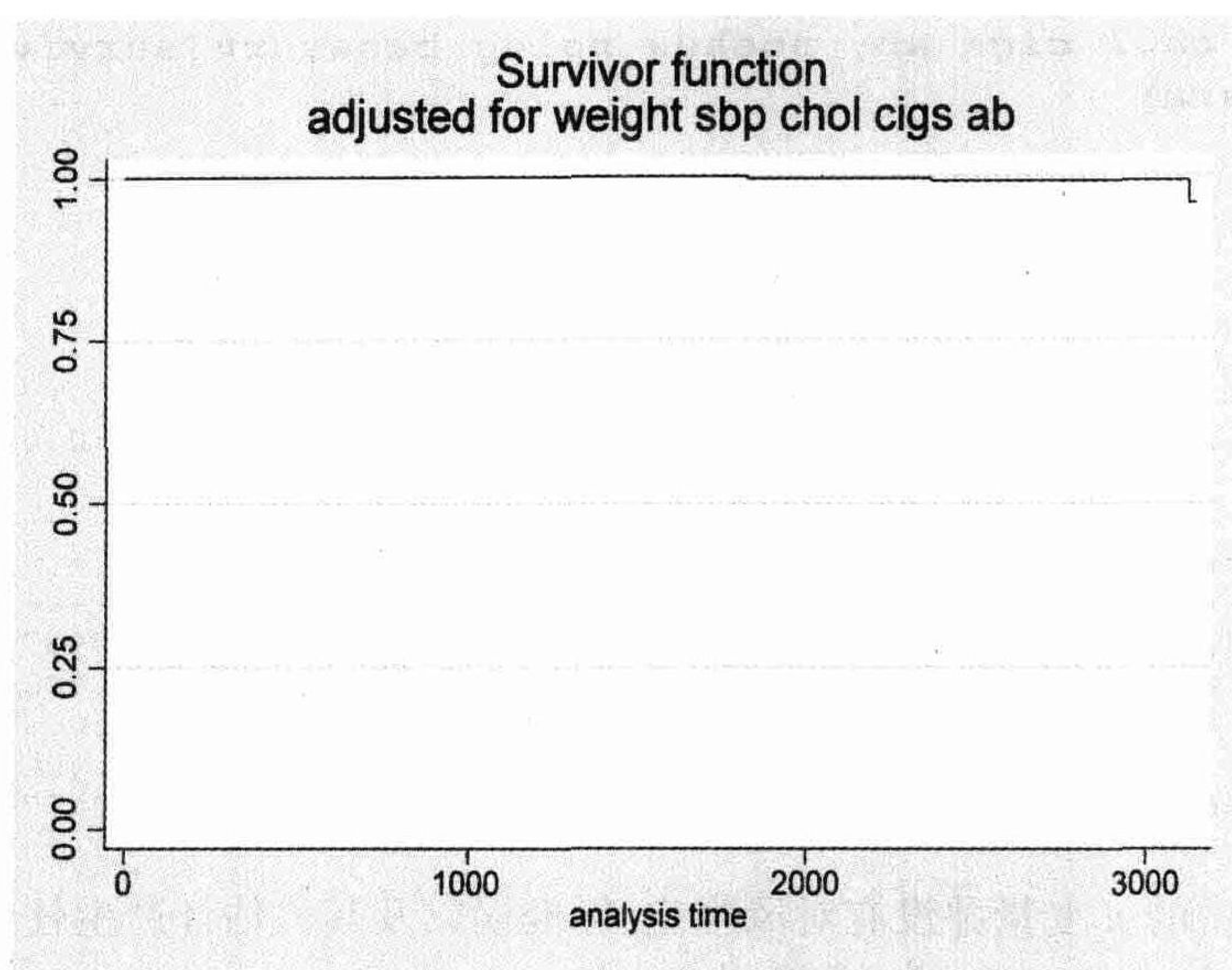


图 11.4

图 11.4 与图 11.3 不同，沿用了通常存活函数的刻度惯例，其纵坐标是从 0 到 1。除在刻度方面不同，图 11.3 和图 11.4 其实描画的是同一曲线。

图 11.5 利用我们由 **stcox** 命令建立的变量 *hazard* 画出了估计的基准累计风险如何随时间而变化。这个图表明，基准累计风险有 8 个上升台阶(因为有 8 名患者“失败”，即得了冠心病)，从接近 0 的水平提高到 0.033。

```
. graph twoway connected hazard time, connect(stairstep) sort
    msymbol(Oh)
```

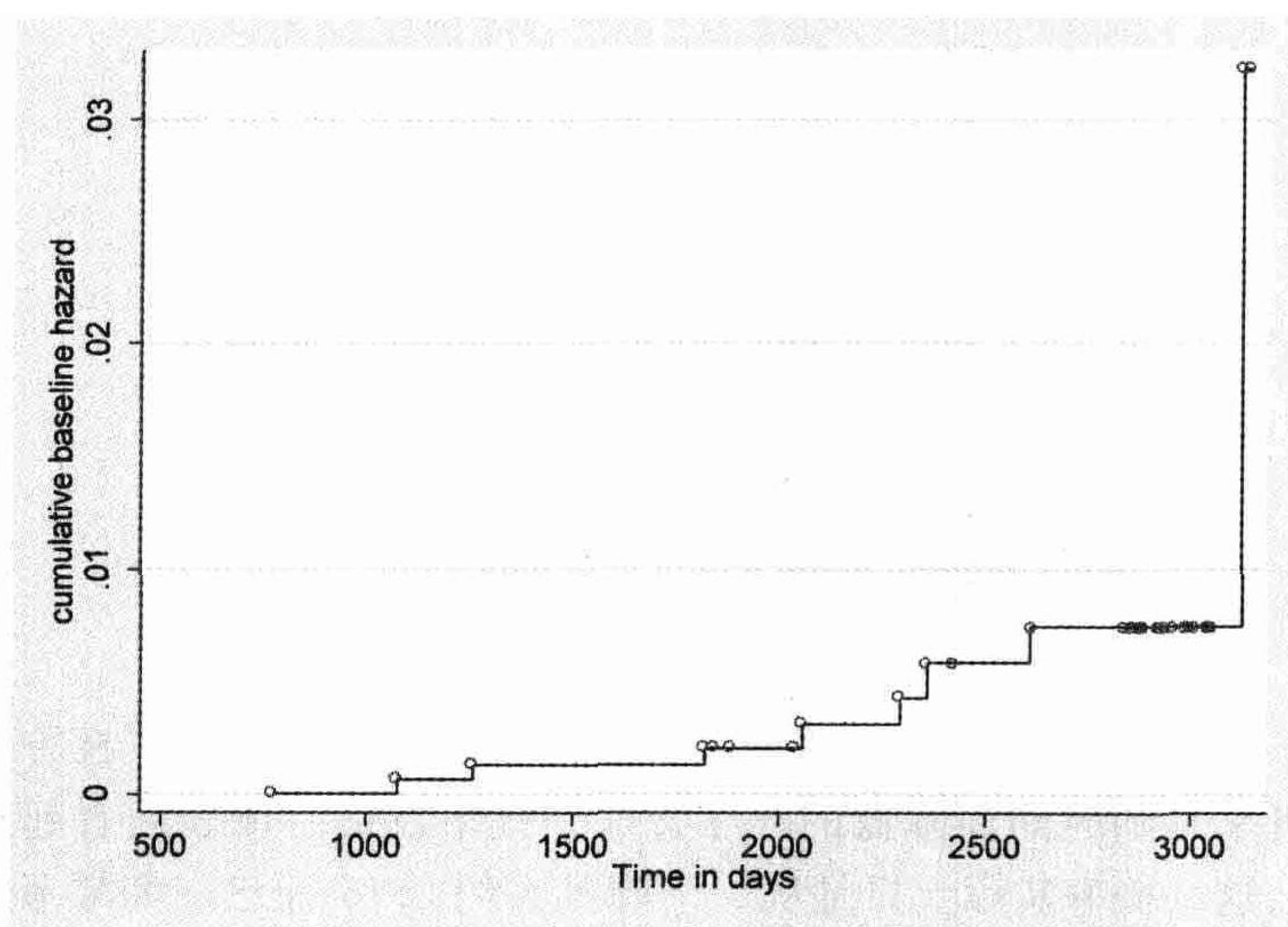


图 11.5

指数回归与 Weibull 回归

Cox 回归是按经验方式来估计基准存活函数，没有参照任何理论分布。另外还有几

种“参数”模型是从假定存活时间服从于一种已知理论分布入手来进行估计的。可以应用的分布类型包括指数(exponential)分布、Weibull 分布、对数正态(lognormal)分布、对数 logistic(log-logistic)分布、Gompertz 分布以及一般化 gamma 分布。基于其中任何一种分布之上的模型都可以使用 **streg** 命令进行拟合。这些模型都有与 Cox 回归同样的一般形式(参见公式[11.2]和[11.3]),但是关于基准风险$h_0(t)$的定义是不同的。本节将用两个例子来加以示范。

如果失败事件的发生是随机的,且风险固定不变,那么存活时间就服从指数分布,并且可以用指数回归(exponential regression)来进行分析。风险不变意味着所研究的个体并不会“老化”,也就是说,他们在观察晚期的失败风险不会比其在观察早期的风险更高或更低。从长期而言,这一假定对于机械和生物都并不合理,但是如果观察期只涉及其生命周期中相对很小一段时,这种假定则大致可以成立。指数模型意味着,其存活函数的对数,即 $\ln(S(t))$,是 t 的线性函数。

第二种常用的参数方法是 Weibull 回归,它的基础是更为一般性的 Weibull 分布。这种方法不要求失败率保持不变,而是允许失败率随时间均匀地提高或降低。Weibull 模型意味着,$\ln(-\ln(S(t)))$,是 $\ln(t)$的线性函数。

图形可以提供关于指数模型或 Weibull 模型恰当性的诊断。比如,再用数据 *aids.dta*,我们先用 Kaplan-Meier 估计取得存活函数 $S(t)$,再制作出一个 $\ln(S(t))$对时间的图形(图 11.6)。图中的 y 轴标签规定为固定的两位数、内含一位小数格式(%2.1f),并且横向排列,以增进该图的可读性。

```
. use aids, clear
(AIDS (Selvin 1995:453))

. sts gen S = S

. generate logS = ln(S)

. graph twoway scatter logS time,
      ylabel(-.8(.1)0, format(%2.1f) angle(horizontal))
```

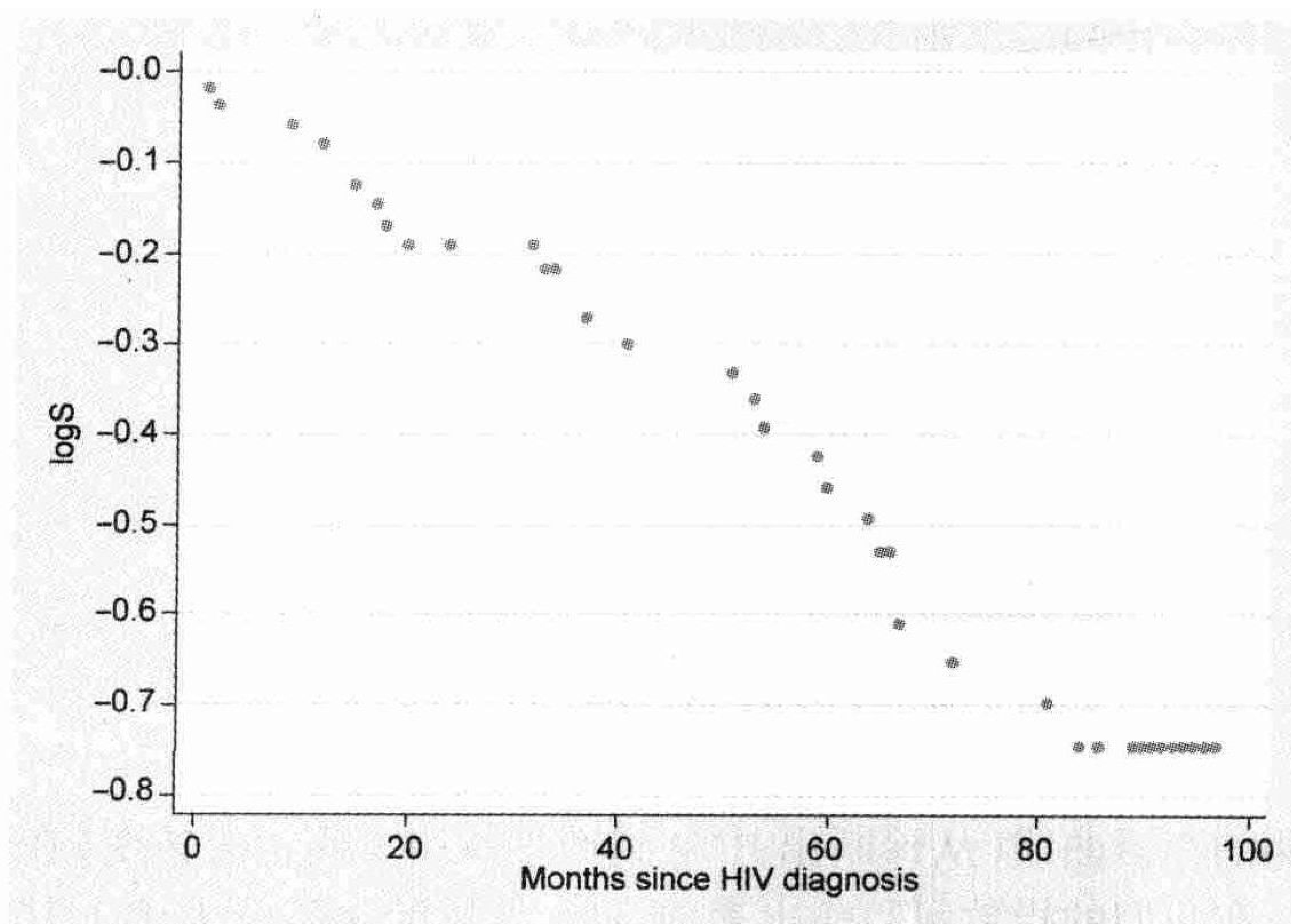


图 11.6

图 11.6 中的模式看起来大致是条直线,这鼓励我们先试一试指数回归:

```
. streg age, dist(exponential) nolog noshow
```

```
Exponential regression -- log relative-hazard form

No. of subjects =           51                Number of obs   =        51
No. of failures =           25
Time at risk    =         3164
                                               LR chi2(1)      =      4.34
Log likelihood  =   -59.996976                 Prob > chi2     =    0.0372

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.074414   .0349626     2.21   0.027     1.008028    1.145172
------------------------------------------------------------------------------
```

通过这一指数回归得到的风险比(1.074)和标准误(0.035)与我们以前用 Cox 回归的相应结果(1.085 和 0.038)并无太大差别。这种相似性反映出经验方法和指数模型所估计的风险函数的对应程度。根据这一指数模型,HIV 阳性的患者发展成 AIDS 的风险每增加一岁便提高 7.4% 。

执行 **streg** 命令以后,**stcurve** 命令可以画出这一模型的累计风险图、或存活函数图、或风险函数图。按照默认设置,**stcurve** 按模型中所有 x 变量取其平均数的条件下画出这些曲线图。我们也可以通过 **at( )** 选项来定义其他的 x 变量值。在数据 *aids.dta* 中,患者的年龄分布于 26 ~50 岁。我们可以通过以下命令来画出在 *age* = 26 岁时的存活函数曲线:

```
. stcurve, surviv at(age=26)
```

使用 **at1( )** 和 **at2( )** 选项还可以在图中提供更多信息,它可以同时显示出在不同 x 取值条件下的存活函数曲线,比如,在最低年龄和最高年龄时的情况:

```
. stcurve, survival at1(age=26) at2(age=50) connect
      (direct direct)
```

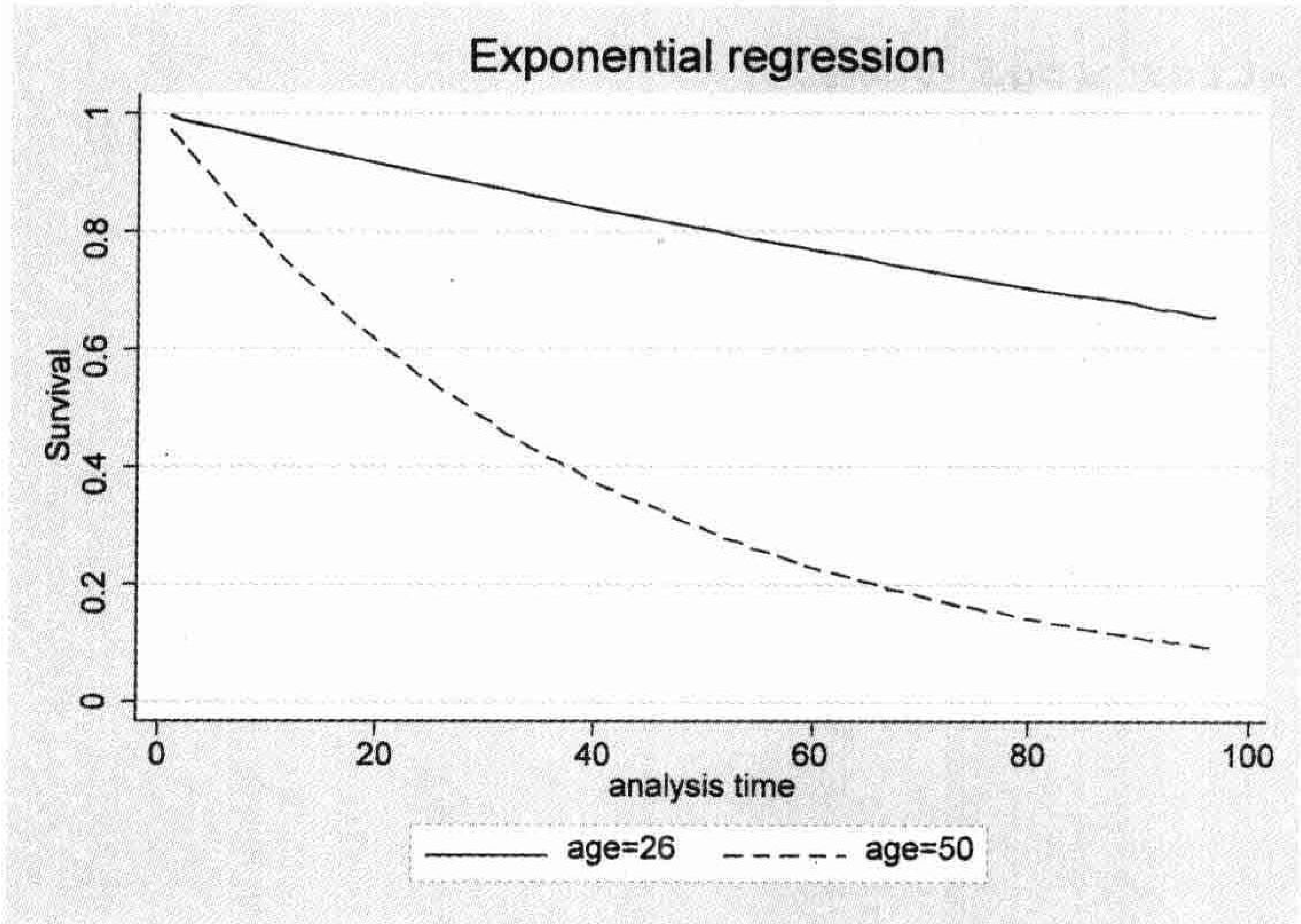


图 11.7

图 11.7 显示出,预测的存活曲线(从诊断出 HIV 到产生 AIDS 症状的转变)在年龄更大的患者中下降更快。在我们的指数回归输出表中 *age* 的风险比显著大于 1 其实表示的是同一情况,但是使用 **stcurve** 加上 **at1( )** 和 **at2( )** 选项值能够对这一效应提供更强大的可视化解释。这些选项在所有三种类型的 **stcurve** 制图中都以同样方式选用:

stcurve, survival　　存活函数

stcurve, hazard 风险函数

stcurve, cumhaz 累计风险函数

除了指数分布以外,**streg** 还可以拟合基于 Weibull 分布的存活模型。Weibull 分布在 ln(*S*(*t*))对 *t* 的图中可能看起来更为曲线化,但是在 ln(- ln(*S*(*t*)))对 ln(*t*)的图中应该是线性的,如图 11.8 所示。而另一方面,指数分布在这两种图形中都显示为直线,并且在 ln(- ln(*S*(*t*)))对 ln(*t*)的图中有斜率等于 1。实际上,图11.8 中的数据点距离斜率为 1 的直线并不远,表明我们前面所做的指数模型就很不错。

```
. generate loglogS = ln(-ln(S))
. generate logtime = ln(time)
. graph twoway scatter loglogS logtime,ylabel(,angle(horizontal))
```

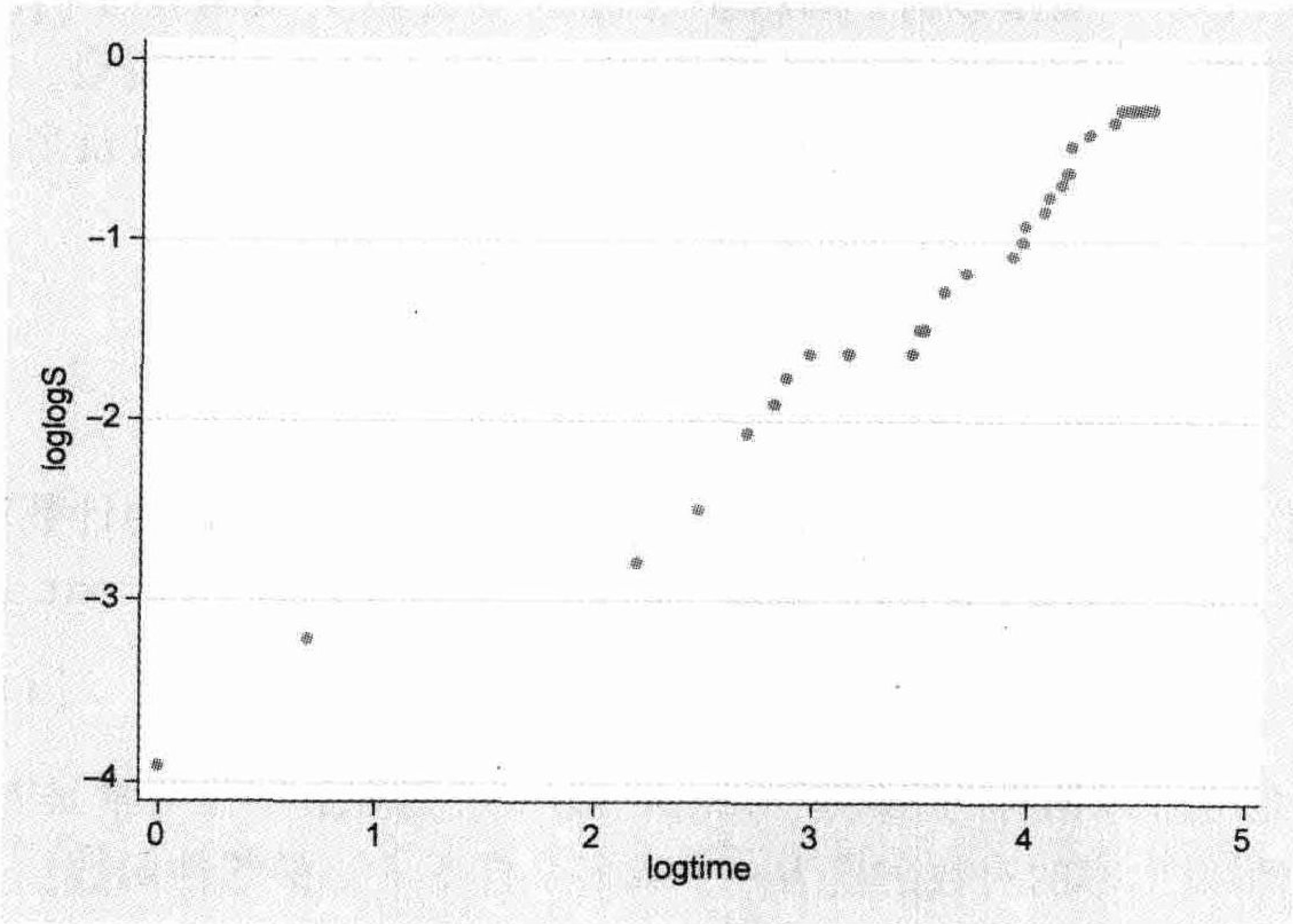


图 11.8

尽管我们在分析这个数据时并不需要像 Weibull 模型那样复杂,但为了示范,下面仍然提供了有关结果。

```
. streg age, dist(weibull) noshow nolog

Weibull regression -- log relative-hazard form

No. of subjects =          51                Number of obs   =         51
No. of failures =          25
Time at risk    =        3164
                                             LR chi2(1)      =       4.68
Log likelihood  =   -59.778257               Prob > chi2     =     0.0306

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.079477   .0363509     2.27   0.023     1.010531    1.153127
-------------+----------------------------------------------------------------
       /ln_p |   .1232638   .1820858     0.68   0.498    -.2336179    .4801454
-------------+----------------------------------------------------------------
           p |   1.131183   .2059723                      .7916643    1.616309
         1/p |   .8840305   .1609694                      .6186934    1.263162
------------------------------------------------------------------------------
```

Weibull 回归取得的风险比估计(1.079)处于我们前面的 Cox 回归和指数回归两者之间。与前面两个模型之间最值得注意的差别是在输出表下部新出现了三行结果。它们都是关于 Weibull 分布形状参数 *p* 的。*p* =1 对应着指数模型,即风险不随时间变

化;$p>1$ 表示风险随时间而增加;$p<1$ 表示风险随时间而减少。p 的95%置信区间为从0.79到1.62,所以在这里我们没有理由拒绝指数模型($p=1$)。Weibull 模型的参数化所关注 ln(p)、p 或 $1/p$ 三个指标尽管看起来不同,然而在数学上却是等价的,所以 Stata 同时提供这三者。在执行 **streg, dist(weibull)** 以后,**stcurve** 可以绘出存活函数、或风险函数、或累计风险函数的曲线,就像在 **streg, dist (exponential)** 或其他 **streg** 模型后一样。

当存活时间实际上服从于指数分布或 Weibull 分布时,指数回归或 Weibull 回归才比 Cox 回归更为可取。当存活时间并不服从这两种分布时,相应这两种模型就是错误设置的,其结果也是误导性的。Cox 回归并不需要对分布形状作任何预先假定,因而它适用于更多的场合。

除了指数模型和 Weibull 模型以外,**streg** 还能拟合更多类型的模型,比如,Gompertz 模型、对数正态模型、对数 logistic 模型或一般化 gamma 分布模型。键入 **help streg** 或者参见《生存分析与流行病学梯度表参考手册》中有关命令以及当前选项的清单。

泊松回归

当事件独立发生且发生概率不变时,那么在给定的一段时期中发生事件的计数就服从于泊松分布(Poisson distribution)。令 r_j 代表发生率(incidence rate):

$$r_j = \frac{\text{事件计数}}{\text{可能发生的事件数}} \qquad [11.4]$$

公式[11.4]中分母的专业术语称为"暴露"(exposure),其测量单位常常是人-年(person-year)。我们将发生率的对数建模为一个或多个自变量 x 的线性函数:

$$\ln(r_t) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \qquad [11.5a]$$

与此等价,这个模型也能描述期望事件计数的对数:

$$\ln(\text{期望计数}) = \ln(\text{暴露}) + \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \qquad [11.5b]$$

假定我们所关注的事件服从泊松过程,那么泊松回归能够求解这些 β 系数的最大似然估计。

奥克雷奇国家实验室关于辐射暴露和癌症死亡的研究数据提供了一个例子。数据 *oakridge.dta* 有56个观测案例,代表了56个年龄与辐射的交互分类(7个年龄类别×8个辐射类别)。对于每一种组合,我们都有相应的死亡数和暴露人年数。

```
Contains data from C:\data\oakridge.dta
  obs:            56                          Radiation (Selvin 1995:474)
 vars:             4                          21 Jul 2005 09:34
 size:           616 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
age             byte   %9.0g       ageg       Age group
rad             byte   %9.0g                  Radiation exposure level
deaths          byte   %9.0g                  Number of deaths
pyears          float  %9.0g                  Person-years
-------------------------------------------------------------------------------
Sorted by:
```

```
. summarize

    Variable |       Obs        Mean    Std. Dev.        Min        Max
-------------+--------------------------------------------------------
         age |        56           4      2.0181          1          7
         rad |        56         4.5    2.312024          1          8
      deaths |        56    1.839286    3.178203          0         16
      pyears |        56    3807.679    10455.91         23      71382

. list in 1/6

     +--------------------------------+
     |   age   rad   deaths   pyears  |
     |--------------------------------|
  1. |  < 45     1        0    29901  |
  2. | 45-49     1        1     6251  |
  3. | 50-54     1        4     5251  |
  4. | 55-59     1        3     4126  |
  5. | 60-64     1        3     2778  |
     |--------------------------------|
  6. | 65-69     1        1     1607  |
     +--------------------------------+
```

那么死亡率是不是随辐射暴露而增加呢？泊松回归发现有统计性显著的影响：

```
. poisson deaths rad, nolog exposure(pyears) irr

Poisson regression                          Number of obs   =         56
                                            LR chi2(1)      =      14.87
                                            Prob > chi2     =     0.0001
Log likelihood =  -169.7364                 Pseudo R2       =     0.0420

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         rad |   1.236469   .0603551     4.35   0.000     1.123657    1.360606
      pyears | (exposure)
------------------------------------------------------------------------------
```

在上述回归中，我们定义死亡事件计数(*deaths*)为因变量，辐射(*rad*)为自变量。泊松回归的"暴露"变量是每个辐射类别的人年数 *pyears*。命令选项 **irr** 要求在结果表中输出发生率比而不是输出回归系数，也就是说，我们直接取得的是 exp(β)估计，而不是默认输出的 β 估计。根据这个发生率比(incidence rate ratio，输出表中为 IRR 列)，当辐射每提高一个类别，死亡率就是原来的 1.236 倍(也就是提高了 23.6％)。尽管这个比值取得了统计显著性，但是模型拟合得并不很好，伪 R^2(参见公式[10.4])只有 0.042。

为了进行拟合优度检验，将泊松模型的预测结果与观测频数加以比较，使用后续命令 **poisgof**：

```
. poisgof

         Goodness-of-fit chi2  =  254.5475
         Prob > chi2(54)       =    0.0000
```

这些拟合优度检验结果(卡方 =254.5，$P<0.000\ 05$)表明，我们的模型预测与实际计数显著不同，从另一个角度再次说明这个模型拟合得不好。

当我们将年龄 *age* 作为第二个自变量纳入模型后，结果就好多了。于是，伪 R^2 提高到 0.596 6，并且拟合优度检验也不再拒绝我们的模型了。

```
. poisson deaths rad age, nolog exposure(pyears) irr

Poisson regression                                Number of obs   =         56
                                                  LR chi2(2)      =     211.41
                                                  Prob > chi2     =     0.0000
Log likelihood =   -71.4653                       Pseudo R2       =     0.5966

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         rad |   1.176673   .0593446     3.23   0.001     1.065924    1.298929
         age |   1.960034   .0997536    13.22   0.000     1.773955    2.165631
      pyears | (exposure)
------------------------------------------------------------------------------

. poisgof

Goodness-of-fit chi2  =  58.00534
Prob > chi2(53)       =    0.2960
```

为了简明,就此我们是将 *rad* 和 *age* 当作连续变量来处理的,并且期望它们对对数死亡率的影响是线性的。但是,实际上这两个自变量的测量都是序次类别。比如,*rad* = 1 代表辐射暴露为 0,*rad* =2 代表 0 ~19 毫西弗特,*rad* =3 代表 20 ~39 毫西弗特,如此等等。当我们寻找非线性关系时可以用另一种方法将暴露类别纳入回归,就是将它们作为一套虚拟变量。下面,我们用 **tabulate** 命令及其 **gen( )** 选项来建立 8 个虚拟变量,*r1* 至 *r8*,来代表 *rad* 的 8 种取值。

```
. tabulate rad, gen(r)

  Radiation |
   exposure |
      level |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          7       12.50       12.50
          2 |          7       12.50       25.00
          3 |          7       12.50       37.50
          4 |          7       12.50       50.00
          5 |          7       12.50       62.50
          6 |          7       12.50       75.00
          7 |          7       12.50       87.50
          8 |          7       12.50      100.00
------------+-----------------------------------
      Total |         56      100.00

. describe

Contains data from C:\data\oakridge.dta
  obs:            56                          Radiation (Selvin 1995:474)
 vars:            12                          21 Jul 2005 09:34
 size:          1064 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
age             byte   %9.0g       ageg       Age group
rad             byte   %9.0g                  Radiation exposure level
deaths          byte   %9.0g                  Number of deaths
pyears          float  %9.0g                  Person-years
r1              byte   %8.0g                  rad==    1.0000
r2              byte   %8.0g                  rad==    2.0000
r3              byte   %8.0g                  rad==    3.0000
r4              byte   %8.0g                  rad==    4.0000
r5              byte   %8.0g                  rad==    5.0000
r6              byte   %8.0g                  rad==    6.0000
r7              byte   %8.0g                  rad==    7.0000
r8              byte   %8.0g                  rad==    8.0000
-------------------------------------------------------------------------------
Sorted by:
```

现在我们将这些虚拟变量中的7个作为回归自变量(省略了其中1个以避免多元共线性)。这一虚拟变量模型额外的复杂性对拟合并没有什么改进。但是,它的确增加了我们的解释。可以看到,辐射对死亡率的总影响主要是来自于最高的两类辐射水平(*r7* 和 *r8*,分别对应着100~119和120及以上毫西弗特两类)。在辐射为这样高水平的组类,死亡率会是无辐射类别的4倍之高。

```
. poisson deaths r2-r8 age, nolog exposure(pyears) irr

Poisson regression                            Number of obs   =         56
                                              LR chi2(8)      =     215.44
                                              Prob > chi2     =     0.0000
Log likelihood = -69.451814                   Pseudo R2       =     0.6080

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          r2 |   1.473591    .426898     1.34   0.181     .8351884    2.599975
          r3 |   1.630688   .6659257     1.20   0.231      .732428    3.630587
          r4 |   2.375967   1.088835     1.89   0.059     .9677429    5.833389
          r5 |   .7278113   .7518255    -0.31   0.758     .0961018    5.511957
          r6 |   1.168477    1.20691     0.15   0.880     .1543195    8.847472
          r7 |   4.433729   3.337738     1.98   0.048     1.013863    19.38915
          r8 |    3.89188   1.640978     3.22   0.001     1.703168    8.893267
         age |   1.961907   .1000652    13.21   0.000     1.775267    2.168169
      pyears | (exposure)
------------------------------------------------------------------------------
```

辐射水平7和8似乎有类似的效应,所以我们可以考虑将它们合并以简化这个模型。首先,我们检验它们的系数是否显著不同。结果它们的差异并不大:

```
. test r7 = r8

( 1)  [deaths]r7 - [deaths]r8 = 0.0

          chi2(  1) =     0.03
        Prob > chi2 =     0.8676
```

然后,再建立一个新的虚拟变量 *r78*,如果 *r7* 和 *r8* 两者之中有一个取值为1时它就等于1,比如:

```
. generate r78 = (r7 | r8)
```

最后,在模型中用这个新自变量取代 *r7* 和 *r8*,比如:

```
. poisson deaths r2-r6 r78 age, irr ex(pyears) nolog

Poisson regression                            Number of obs   =         56
                                              LR chi2(7)      =     215.41
                                              Prob > chi2     =     0.0000
Log likelihood = -69.465332                   Pseudo R2       =     0.6079

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          r2 |   1.473602   .4269013     1.34   0.181     .8351949    2.599996
          r3 |   1.630718   .6659381     1.20   0.231     .7324415    3.630655
          r4 |   2.376065    1.08888     1.89   0.059     .9677823    5.833629
          r5 |   .7278387   .7518538    -0.31   0.758     .0961055    5.512165
          r6 |   1.168507   1.206942     0.15   0.880     .1543236    8.847704
         r78 |   3.980326   1.580024     3.48   0.001     1.828214    8.665833
         age |   1.961722    .100043    13.21   0.000     1.775122    2.167937
      pyears | (exposure)
------------------------------------------------------------------------------
```

我们还可以照这种方式继续简化这个模型。进行每一步时,**test** 都有助于评价将两个虚拟变量合并是否合理。

一般化线性模型

一般化线性模型(GLM)的形式如下:

$$g[E(y)] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k, \quad y \sim F \qquad [11.6]$$

其中 $g[\ ]$ 是连接函数(link function),F 代表分布家族。这一通用公式包含了许多特殊模型。比如,$g[\ ]$ 是恒等函数,且 y 服从于正态(高斯)分布,我们就得到线性回归模型:

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k, \quad y \sim Normal \qquad [11.7]$$

如果 $g[\ ]$ 是 logit 函数且 y 服从于贝努里(Bernoulli)分布,我们就得到 logit 回归模型:

$$\text{logit}[E(y)] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k, \quad y \sim Bernoulli \qquad [11.8]$$

由于其广泛的应用,GLM 在本书中多处地方本来都可以加以介绍。它与本章的关联是因为其拟合事件模型的能力。比如,泊松回归要求 $g[\ ]$ 是自然对数函数并且 y 服从泊松(Poisson)分布:

$$\ln[E(y)] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k, \quad y \sim Poisson \qquad [11.9]$$

正如对这样一个灵活方法的期盼,Stata 的 **glm** 命令容纳许多不同的选项。用户不仅可以指定分布家族和连接函数,而且可以指定方差估计方法、拟合程序、输出和补偿方面的细节。这些选项使 **glm** 即使用于已经有专门命令的那些模型(如 **regress**、**logistic** 或 **poisson**)时也是很有用的替代方法。

我们可以展示一个"一般"的 **glm** 命令如下:

```
. glm y x1 x2 x3, family(familyname) link(linkname)
      lnoffset(exposure) eform jknife
```

其中,**family**( )指定了 y 的分布家族,**link**( )指定连接函数,**lnoffset**( )用于对"暴露"变量的补偿(offset),就像在泊松回归所需的那样。选项 **eform** 要求回归系数为指数形式的 exp(β)而不是 β。而 **jknife** 选项则是指定标准误要通过刀切法(jackknife)来加以计算。

可用的分布家族有:

family(gaussian)	高斯分布,即正态分布(默认)
family(igaussian)	反高斯分布(inverse Gaussian)
family(binomial)	贝努里二项分布(Bernoulli binomial)
family(poisson)	泊松分布(Poisson)
family(nbinomial)	负二项分布(negative binomial)
family(gamma)	Gamma 分布

我们还可以指定一个数字或变量表示二项分布分母 N(试验次数),或者用一个数字表示负二项方差和偏差度函数(deviance function),这些就通过在 **family**( ) 选项中对它们加以声明:

family(binomial #)

family(binomial *varname*)

family(nbinomial #)

可用的连接函数有:

link(identity) 恒等函数(默认)
link(log) 对数函数
link(logit) logit 函数
link(probit) probit 函数
link(cloglog) 补充双对数函数
link(opower #) 发生比幂函数
link(power #) 幂函数
link(nbinomial) 负二项函数
link(loglog) 双对数函数
link(logc) 对数余角函数

系数的方差或标准误可以通过各种方法来进行估计。以下列出部分 **glm** 的方差估计选项:

opg Berndt、Hall、Hall 和 Hausman 的"B-H-cubed"方差估计
oim 观测信息矩阵方差估计
robust Huber/White/三明治方差估计
unbiased 无偏三明治方差估计
nwest 异方差性和自相关一致性方差估计
jknife 刀切法方差估计
jknife1 一步刀切法方差估计
bstrap 自助法方差估计。默认为 199 次重复,指定其他数字时可加 **bsrep(#)** 选项

要想取得各种选项的完整清单及有关技术细节,请查询《基础参考手册》中的 **glm**。更深入的 GLM 处理可以参阅 Hardin 和 Hilbe(2001)的著作。

第 6 章开始于用美国 50 个州及哥伦比亚特区学生平均支出(*expense*)与平均综合 SAT 分(*csat*)的数据(*states.dta*)做了一个简单回归:

```
. regress csat expense
```

我们也可以用以下命令来拟合同一个模型,并且取得完全同样的估计:

```
. glm csat expense, link(identity) family(gaussian)
Iteration 0:   log likelihood = -279.99869

Generalized linear models                    No. of obs      =        51
Optimization     : ML: Newton-Raphson        Residual df     =        49
                                             Scale param     =  3577.678
Deviance         =  175306.2097              (1/df) Deviance =  3577.678
Pearson          =  175306.2097              (1/df) Pearson  =  3577.678

Variance function: V(u) = 1                  [Gaussian]
Link function    : g(u) = u                  [Identity]
Standard errors  : OIM

Log likelihood   = -279.9986936              AIC             =  11.05877
BIC              =   175298.346

------------------------------------------------------------------------------
        csat |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |  -.0222756    .0060371    -3.69   0.000    -.0341082   -.0104431
       _cons |   1060.732     32.7009    32.44   0.000     996.6399    1124.825
------------------------------------------------------------------------------
```

因为 **link(identity)** 和 **family(gaussian)** 都是默认选项,我们在上一个 **glm** 命令其实用不着它们。

实际上,**glm** 命令能做的工作比重复 **regress** 结果要多得多。比如,我们可以拟合同一个 OLS 模型,但是取得自助法计算的标准误[12]:

```
. glm csat expense, link(identity) family(gaussian) bstrap

Iteration 0:   log likelihood = -279.99869

Bootstrap iterations (199)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
.................................................

Generalized linear models                          No. of obs      =        51
Optimization     : ML: Newton-Raphson              Residual df     =        49
                                                   Scale param     =  4124.656
Deviance         =  175306.2097                    (1/df) Deviance =  3577.678
Pearson          =  175306.2097                    (1/df) Pearson  =  3577.678

Variance function: V(u) = 1                        [Gaussian]
Link function    : g(u) = u                        [Identity]
Standard errors  : Bootstrap

Log likelihood   = -279.9986936                    AIC             =  11.05877
BIC              =   175298.346

------------------------------------------------------------------------------
             |              Bootstrap
        csat |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     expense |  -.0222756   .0039284    -5.67   0.000    -.0299751   -.0145762
       _cons |   1060.732   25.36566    41.82   0.000     1011.017    1110.448
------------------------------------------------------------------------------
```

自助法标准误是通过在规模为 $n=51$ 的原始数据中所做的 199 次 $n=51$ 的重置随机抽样来反映系数估计的观测变异的。在此例中,自助法标准误小于相应理论标准误,并且导致置信区间也比较窄。

与此类似,我们能用 **glm** 来重复第 10 章中的第一个 **logistic** 回归。在以下示例中,我们要求输出刀切法标准误和优势比,即指数形式(**eform**)的系数:

```
. glm any date, link(logit) family(bernoulli) eform jknife

Iteration 0:   log likelihood = -12.995268
Iteration 1:   log likelihood = -12.991098
Iteration 2:   log likelihood = -12.991096

Jackknife iterations (23)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
.......................

Generalized linear models                          No. of obs      =        23
Optimization     : ML: Newton-Raphson              Residual df     =        21
                                                   Scale param     =         1
Deviance         =  25.98219269                    (1/df) Deviance =  1.237247
Pearson          =   22.8885488                    (1/df) Pearson  =  1.089931

Variance function: V(u) = u*(1-u)                  [Bernoulli]
Link function    : g(u) = ln(u/(1-u))              [Logit]
Standard errors  : Jackknife
```

[12]【译注:Stata 第 9 版现输出格式和 BIC、自助法标准误、z、95% CI 的计算值都略有变化。】

```
Log likelihood     = -12.99109634                    AIC              =  1.303574
BIC                =  19.71120426

------------------------------------------------------------------------------
             |               Jackknife
         any | Odds Ratio   Std. Err.       z    P>|z|      [95% Conf. Interval]
-------------+----------------------------------------------------------------
        date |   1.002093   .0015486     1.35   0.176      .9990623    1.005133
------------------------------------------------------------------------------
```

[13]

而本章最后的 **poisson** 回归则是对应着以下这个 **glm** 模型：

```
. glm deaths r2-r6 r78 age, link(log) family(poisson)
     lnoffset(pyears) eform
```

尽管 **glm** 能够复制由许多专门命令所拟合的模型，并且还添加了一些新功能，那些专门命令仍有它们自己的优势，包括在速度方面以及在可供用户决定的选项方面。**glm** 最独特的吸引力在于它有能力去拟合 Stata 并无专门命令的那些类型的模型。

[13]【译注：Stata 第 9 版现输出格式和 BIC 计算值略有变化。】

12 主成分、因子和聚类分析[14]

主成分(principal components)和因子分析(factor analysis)都是将许多相关的变量合并成少数几个潜在维度(underlying dimensions),因而提供了用于简化(simplification)的方法。为了达到简化的目的,分析人员必须从诸多不同种类的备选方法中进行选择。如果数据的确反映了明确的数个潜在维度,那么不同方法可能会收敛于类似的结果上。但是,当不存在明确的潜在维度的情况下,不同方法得到的结果往往会出现分歧。对这些方法的试验能够告诉我们一个特定的结果的稳定性如何,或者它在多大程度上取决于特定分析技术的人为选择。

Stata 采用五条基本命令来实现主成分和因子分析:

pca 主成分分析(principle components analysis)。

factor 提取若干不同类型的因子。

greigen 根据最近的 **pca** 或 **factor** 构建碎石图(scree graph)(即特征值标绘图,plot of the eigenvalues)。

rotate 在执行 **factor** 后,进行正交(即相互独立的因子)或斜交(即因子不相互独立)的旋转。

predict 在执行 **pca** 、**factor** 或 **rotate** 之后,创建因子分(factor scores)(即复合变量,composite variables)和其他的案例统计量。

由 **predict** 创建的复合变量随后可被像任何其他 Stata 变量一样加以保存、列出、制图和分析。

对于那些采用加总其他变量的老方法而不是用因子分析来创建复合变量的用户,可以通过计算一个 α 信度系数(reliability coefficient)对所得结果进行评价:

alpha Cronbach 的 α 信度。

聚类分析(cluster analysis)则不是将变量加以合并,而是通过找到非重叠的、基于经验的数个类型或组别将观测案例加以合并。聚类分析方法甚至比因子分析更为多样化,但理论却更少。Stata 的 **cluster** 命令可以进行聚类分析、结果制图以及形成区分结果组别的新变量等工作。

本章所描述的方法可以通过以下菜单方式来操作:

Statistics-Multivariate analysis 多元分析

Graphics-More statistical graphs 更多统计图形

---

[14]【译注:翻译中发现原书本章中的命令与输出因 Stata 9.0 更新而有明显出入,因此由原作者对本章作了修订,本章依据这一修订文稿翻译。】

命令示范

```
. pca x1 -x20
```
对变量 *x1* 到 *x20* 进行主成分分析。

```
. pca x1 -x20 , mineigen (1)
```
对变量 *x1* 到 *x20* 进行主成分分析，保留特征值大于 1 的成分。

```
. factor x1 -x20 , ml factor (5)
```
采用最大似然法对变量 *x1* 到 *x20* 进行因子分析，只保留前五个因子。

```
. screeplot
```
画出由最近的 **factor** 命令得到的特征值对因子或成分数目的碎石图或图形。

```
. rotate, varimax factors (2)
```
对由最近的 **factor** 命令得到的前两个因子进行正交（用方差最大法，varimax）旋转。

```
. rotate, promax factors (3)
```
对由最近的 **factor** 命令得到的前三个因子进行斜交（用幂方法，promax）旋转。

```
. predict f1 f2 f3
```
基于最近的 **factor** 和 **rotate** 命令，创建三个新的名为 *f1*、*f2* 和 *f3* 的因子分变量（factor score variables）。

```
. alpha x1 -x10
```
计算作为 *x1* 到 *x10* 的合计的复合变量的 Cronbach 的 α 信度系数。以负值方式输入项目的含义通常是反向的。可以通过一些选项来取消这一默认设置，或者以原始变量的合计或以其标准化变量的合计来构成复合变量。

```
. cluster centroidlinkage x y z w, L2 name (L2 cent)
```
使用变量 *x*、*y* 和 *z* 以重心法（centroid linkage）进行凝聚式（agglomerative）聚类分析。欧氏距离（Euclidean distance）（**L2**）测量了观测案例之间的相异性（dissimilarity）。这一聚类分析的结果被以 *L2 cent* 的名称加以保存。

```
. cluster dendrogram, ylabel(0(.5)3) cutnumber (20)
  xlabel (,angle (vertical))
```
画出前次聚类分析结果的树状图（tree graph）或系统树图（dendrogram）。**cutnumber(20)** 设定将一些最为相似的观测案例聚合之后，图形中只保留 20 个聚类。标签以紧凑的垂直方式显示在图形的下面。**cluster dendrogram** 的作用和 **clsuter tree** 完全一样。

```
. cluster generate ctype =groups (3), name (L2cent)
```
创建一个新变量 *ctype*（取值为 1、2 或 3），通过该变量将每条观测案例按名为 *L2 cent* 的聚类分析结果归类到其前三个组别中去。

主成分

我们将使用描述太阳系九颗主要行星的一个小规模数据集 *planets.dta*（取自 Beatty 等，1981）来举例说明基本的主成分和因子分析命令。该数据包含了以原始数

据及其自然对数形式保存的几个变量。这里,采用对数是为了消除偏态、并将变量之间的关系线性化。

```
Contains data from C:\data\planets.dta
  obs:             9                          Solar system data
 vars:            12                          22 Jul 2005 09:49
 size:           441 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
planet          str7   %9s                    Planet
dsun            float  %9.0g                  Mean dist. sun, km*10^6
radius          float  %9.0g                  Equatorial radius in km
rings           byte   %8.0g       ringlbl    Has rings?
moons           byte   %8.0g                  Number of known moons
mass            float  %9.0g                  Mass in kilograms
density         float  %9.0g                  Mean density, g/cm^3
logdsun         float  %9.0g                  natural log dsun
lograd          float  %9.0g                  natural log radius
logmoons        float  %9.0g                  natural log (moons + 1)
logmass         float  %9.0g                  natural log mass
logdense        float  %9.0g                  natural log dense
-------------------------------------------------------------------------------
Sorted by:  dsun
```

为了提取初始因子和主成分,请使用 **factor** 命令及其后跟随的变量清单(变量顺序任意)、以及以下选项之一:

pcf　　主成分因子法(principical component factoring)

pf　　主因子法(principal factoring)(默认选项)

ipf　　使用迭代公因子方差(iterated communalities)的主因子法

ml　　最大似然因子法(maximum-likelihood factoring)

主成分通过专门的命令 **pca** 计算得到。请键入 **help pca** 或 **help factor** 查看这些命令的选项。

为了取得主成分因子,请键入:

. factor *rings logdsun - logdense*, *pcf*

(obs =9)

```
Factor analysis/correlation                    Number of obs    =         9
    Method: principal-component factors       Retained factors =         2
    Rotation: (unrotated)                      Number of params =        11

    --------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+------------------------------------------------------------
        Factor1  |      4.62365      3.45469            0.7706       0.7706
        Factor2  |      1.16896      1.05664            0.1948       0.9654
        Factor3  |      0.11232      0.05395            0.0187       0.9842
        Factor4  |      0.05837      0.02174            0.0097       0.9939
        Factor5  |      0.03663      0.03657            0.0061       1.0000
        Factor6  |      0.00006                         0.0000       1.0000
    --------------------------------------------------------------------------
    LR test: independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 = 0.0000

Factor loadings (pattern matrix) and unique variances

    -------------------------------------------------
        Variable |  Factor1   Factor2 |   Uniqueness
    -------------+--------------------+--------------
           rings |   0.9792    0.0772 |      0.0353
         logdsun |   0.6710   -0.7109 |      0.0443
          lograd |   0.9229    0.3736 |      0.0088
        logmoons |   0.9765    0.0003 |      0.0465
         logmass |   0.8338    0.5446 |      0.0082
        logdense |  -0.8451    0.4705 |      0.0644
    -------------------------------------------------
```

只有前两个成分具有大于1的特征值(eigenvalue),同时这两个成分解释了六个变量组合方差(combined variance)的96%还多。不重要的第3到第6个主成分在随后的分析中可以放心地被省略。

两个 **factor** 选项用于控制提取因子的数目:

factor(#) 这里的#设定因子数目

mineigen(#) 这里的#设定被保留因子的最小特征值

主成分因子法(**pcf**)程序自动删除那些特征值小于1的因子,因此

```
.factor rings logdsun - logdense, pcf
```

等价于

```
. factor rings logdsun - logdense, pcf mineigen (1)
```

在本例中,如果键入以下命令的话,我们也将得到同样的结果,

```
. factor rings logdsun - logdense, pcf factors (2)
```

要想在每次 **factor** 之后查看碎石图(特征值对主成分或因子数目的标绘图),可使用 **screeplot** 命令[15]。图12.1中位于特征值等于1处的水平线标示了保留主成分的常用分界点(cutoff),同时再次强调了本例中的成分3到6并不重要。

```
. greigen, yline (1)
```

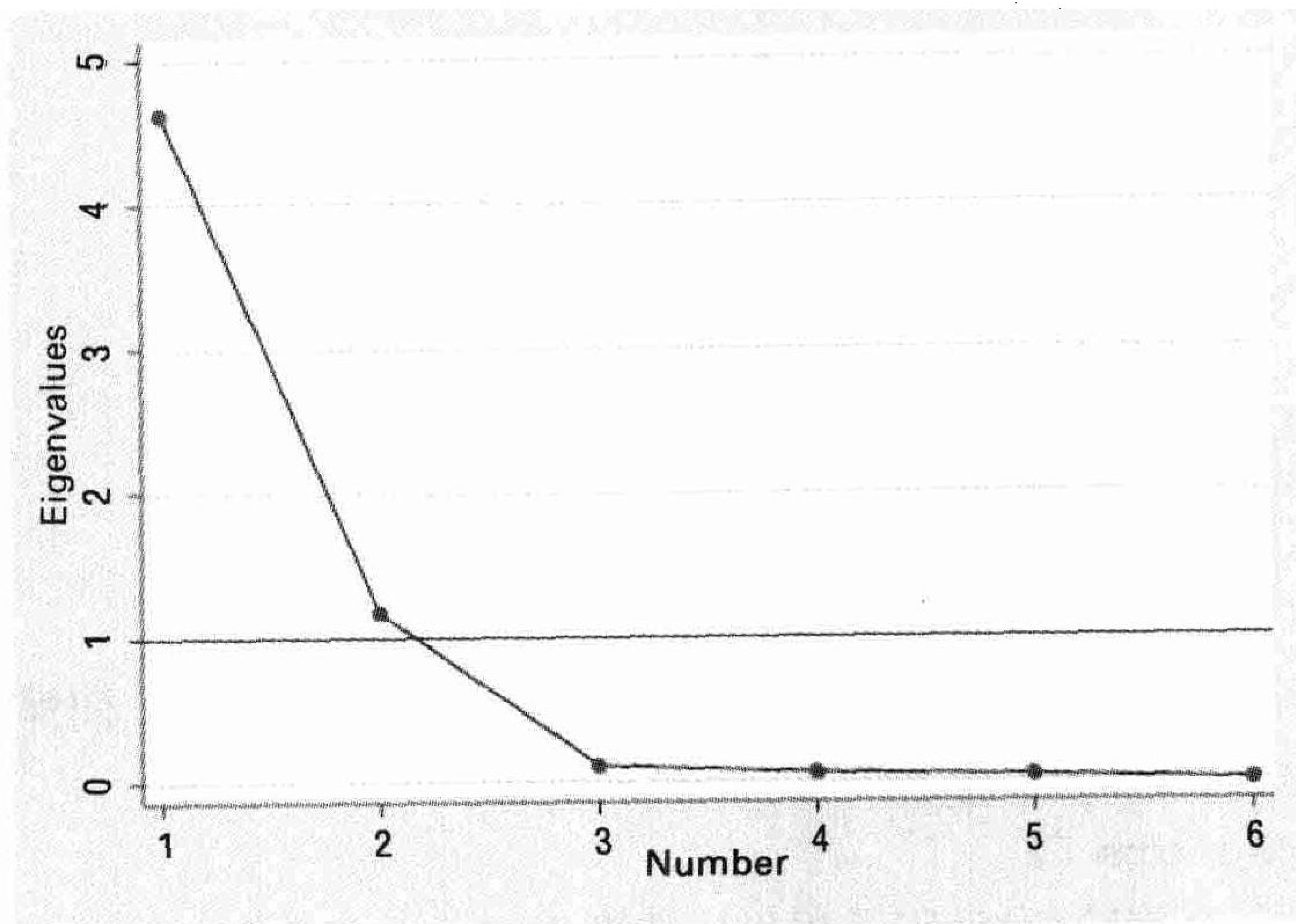


图12.1

旋 转

旋转(rotation)会进一步简化因子结构。在提取因子之后,键入 **rotate** 命令以及以下这些选项之一:

varimax 最大方差正交旋转,适用于相互独立的因子或成分(默认选项)。

promax( ) promax 斜交旋转,允许因子或成分之间相关。选择一个小于等于4的数(promax 的指数);数越大,因子间的相关程度越高。**promax(3)** 为默认设定。

另外还有两个 **rotate** 的选项:

---

[15]【译注:此处也可以用 **greigen** 命令,它与 **screeplot** 等价(见下例)。】

factors( )　　这一选项与和 **factor** 搭配时一样，也是设定要保留多少个因子。

entropy　　最小信息熵正交旋转。

无论应用 **pcf**、**pf**、**ipf** 还是 **ml** 选项中的哪一类方法来做因子分析，之后都能进行旋转。在本节，我们将一直基于我们的 **pcf** 例子。要对行星数据中发现的前两个成分作正交旋转（默认的旋转方法），我们键入：

. rotate

```
Factor analysis/correlation                        Number of obs    =        9
    Method: principal-component factors            Retained factors =        2
    Rotation: orthogonal varimax (Kaiser off)      Number of params =       11

    --------------------------------------------------------------------------
         Factor  |     Variance   Difference        Proportion   Cumulative
    -------------+------------------------------------------------------------
        Factor1  |      3.36900      0.94539            0.5615       0.5615
        Factor2  |      2.42361                         0.4039       0.9654
                 |
    --------------------------------------------------------------------------
    LR test:  independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 = 0.0000

Rotated factor loadings (pattern matrix) and unique variances

    ---------------------------------------------------
        Variable |  Factor1   Factor2 |   Uniqueness
    -------------+--------------------+--------------
           rings |   0.8279    0.5285 |      0.0353
         logdsun |   0.1071    0.9717 |      0.0443
          lograd |   0.9616    0.2580 |      0.0088
        logmoons |   0.7794    0.5882 |      0.0465
         logmass |   0.9936    0.0678 |      0.0082
        logdense |  -0.3909   -0.8848 |      0.0644
    ---------------------------------------------------

Factor rotation matrix

    ---------------------------------
                 |  Factor1  Factor2
    -------------+-------------------
         Factor1 |   0.7980
         Factor2 |   0.6026  -0.7980
    ---------------------------------
```

本例采纳了所有的默认设定：最大方差法旋转和保留与最近一次 **factor** 同样的因子数目。采用以下命令，我们可以明确地要求进行同样的旋转：

. rotate, varimax factors (2)

对于最近一次提取因子的斜交旋转（允许因子相关），请键入：

. rotate, promax

```
Factor analysis/correlation                        Number of obs    =        9
    Method: principal-component factors            Retained factors =        2
    Rotation: oblique promax (Kaiser off)          Number of params =       11

    --------------------------------------------------------------------------
         Factor  |     Variance   Proportion    Rotated factors are correlated
    -------------+------------------------------------------------------------
        Factor1  |      4.12467       0.6874
        Factor2  |      3.32370       0.5539
    --------------------------------------------------------------------------
    LR test:  independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 = 0.0000

Rotated factor loadings (pattern matrix) and unique variances
```

Variable	Factor1	Factor2	Uniqueness
rings	0.7626	0.3466	0.0353
logdsun	-0.1727	1.0520	0.0443
lograd	0.9926	0.0060	0.0088
logmoons	0.6907	0.4275	0.0465
logmass	1.0853	-0.2154	0.0082
logdense	-0.1692	-0.8719	0.0644

Factor rotation matrix

	Factor1	Factor2
Factor1	0.9250	0.7898
Factor2	0.3800	-0.6134

默认状态下,本例使用的 promax 的指数为 3。我们可以明确设定 promax 指数和想要得到的因子数:

```
. rotate, promax (3) factors (2)
```

promax(4)将允许对负载矩阵(loading matrix)作进一步的简化,但是将以更强的因子间相关和更低的总方差解释比例作为代价。

进行 promax 旋转之后,*rings*、*lograd*、*logmoons* 和 *logmass* 在因子 2 上的负载最高。这看起来是一个"大规模/多卫星"维度。*logdsun* 和 *logdense* 在因子 1 上的负载更高了,构成了一个"遥远/低密度"维度,下一节将展示如何创建代表这些维度的新变量。一个后续因子分析的制图命令 **loadingplot** 有助于将其可视化。

```
. loadingplot, factors (2) yline(0) xline (0)
```

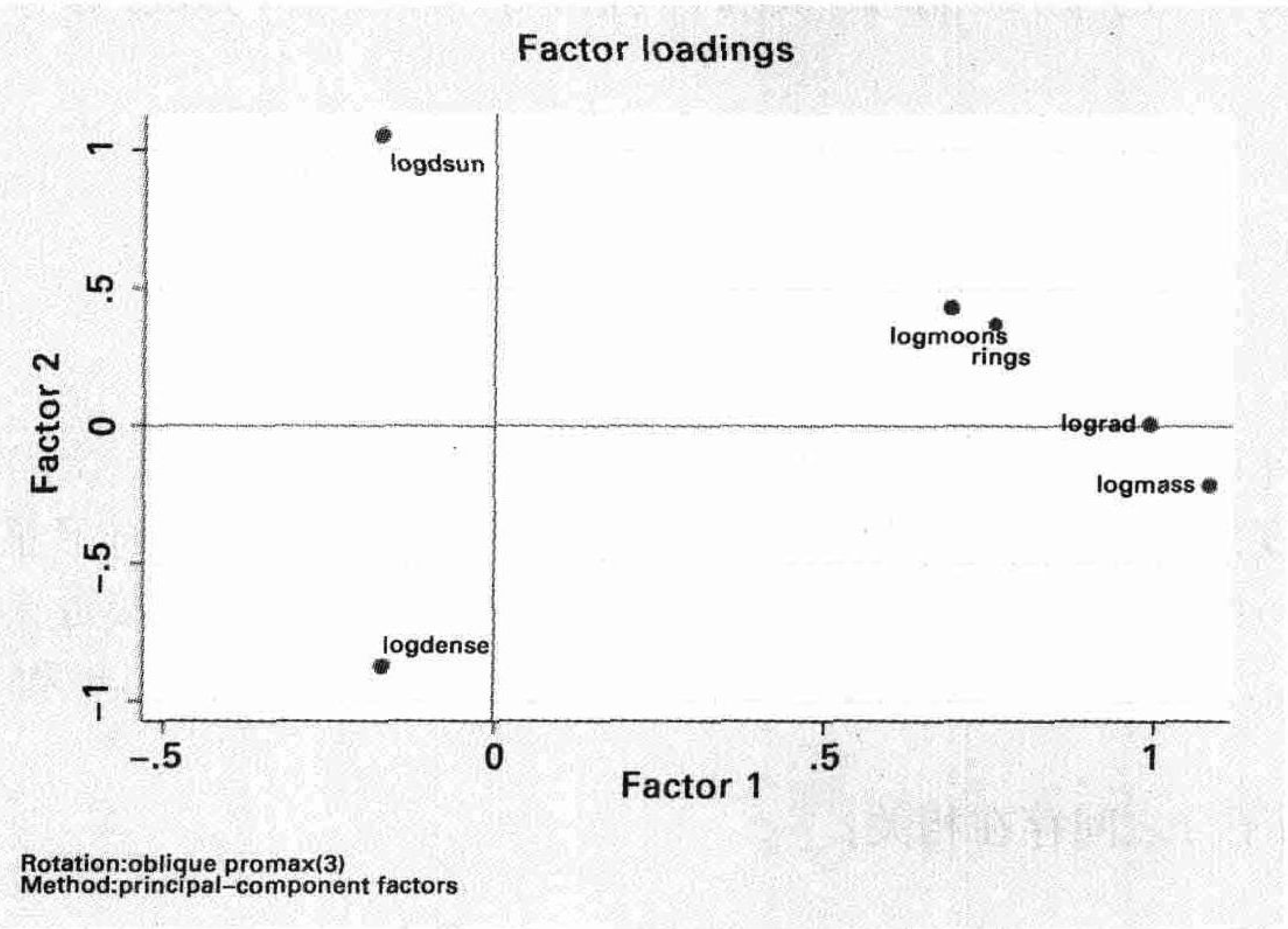


图 12.2

因子分

因子分(factor scores)是通过将每个变量标准化为平均数等于 0 和方差等于 1、然后以因子分系数进行加权合计为每个因子构成的线性组合(linear composites)。基于最近的 **rotate** 或 **factor** 结果,**predict** 会自动进行这些计算。在 **predict** 命令中,我们提供了新变量的名称,比如,*f1* 和 *f2*。

```
. predict f1 f2
(regression scoring assumed)
Scoring coefficients (method = regression; based on promax(3) rotated factors)

    -----------------------------------
        Variable |   Factor1    Factor2
    -------------+---------------------
           rings |   0.22099    0.12674
         logdsun |  -0.09689    0.48769
          lograd |   0.30608   -0.03840
        logmoons |   0.19543    0.16664
         logmass |   0.34386   -0.14338
        logdense |  -0.01609   -0.39127
    -----------------------------------

. label variable f1 "Large size /many satellites"
. label variable f2 "Far out /low density"
. list planet f1 f2

     +--------------------------------+
     |  planet          f1         f2 |
     |--------------------------------|
  1. | Mercury   -.9172388  -1.256881 |
  2. |   Venus   -.5160229  -1.188757 |
  3. |   Earth   -.3939372  -1.035242 |
  4. |    Mars   -.6799535  -.5970106 |
  5. | Jupiter    1.342658   .3841085 |
     |--------------------------------|
  6. |  Saturn    1.184475   .9259058 |
  7. |  Uranus    .7682409   .9347457 |
  8. | Neptune     .647119   .8161058 |
  9. |   Pluto    -1.43534   1.017025 |
     +--------------------------------+
```

作为标准化的变量,新的因子分 f1 和 f2 具有(近似)等于 0 的平均数和等于 1 的标准差:

```
. summarize f1 f2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          f1 |         9   -3.31e-09           1   -1.43534   1.342658
          f2 |         9    9.93e-09           1  -1.256881   1.017025
```

因此,因子分是以距离其平均数的标准差单位进行测量的。比如,水星(Mercury)低于大规模/多卫星(f1)维度的平均数大约 0.92 个标准差,因为它很小而且没有卫星。水星低于遥远/低密度(f2)维度的平均数大约 1.26 个标准差,因为它实际上接近太阳而且具有高密度。相比而言,土星(Saturn)高于这两个维度平均数分别为 1.18 和 0.93 个标准差。

promax 旋转允许因子分之间存在相关:

```
. correlate f1 f2
(obs =9)

             |       f1       f2
-------------+------------------
          f1 |   1.0000
          f2 |   0.4974   1.0000
```

因子 1 上的得分与因子 2 上的得分之间具有中度正相关:遥远/低密度行星也更可能是具有许多卫星的更大行星。

另一个后因子分析制图命令,**scoreplot** 可绘出这些观测案例的因子分的散点图。当作主成分使用的话,因子可以帮助识别多元特异值,或由远离大多数案例的那些案例

所形成的聚类。图 12.3 显示了三个不同类型的行星。

```
. scoreplot, mlabel(planet) yline (0) xline (0)
```

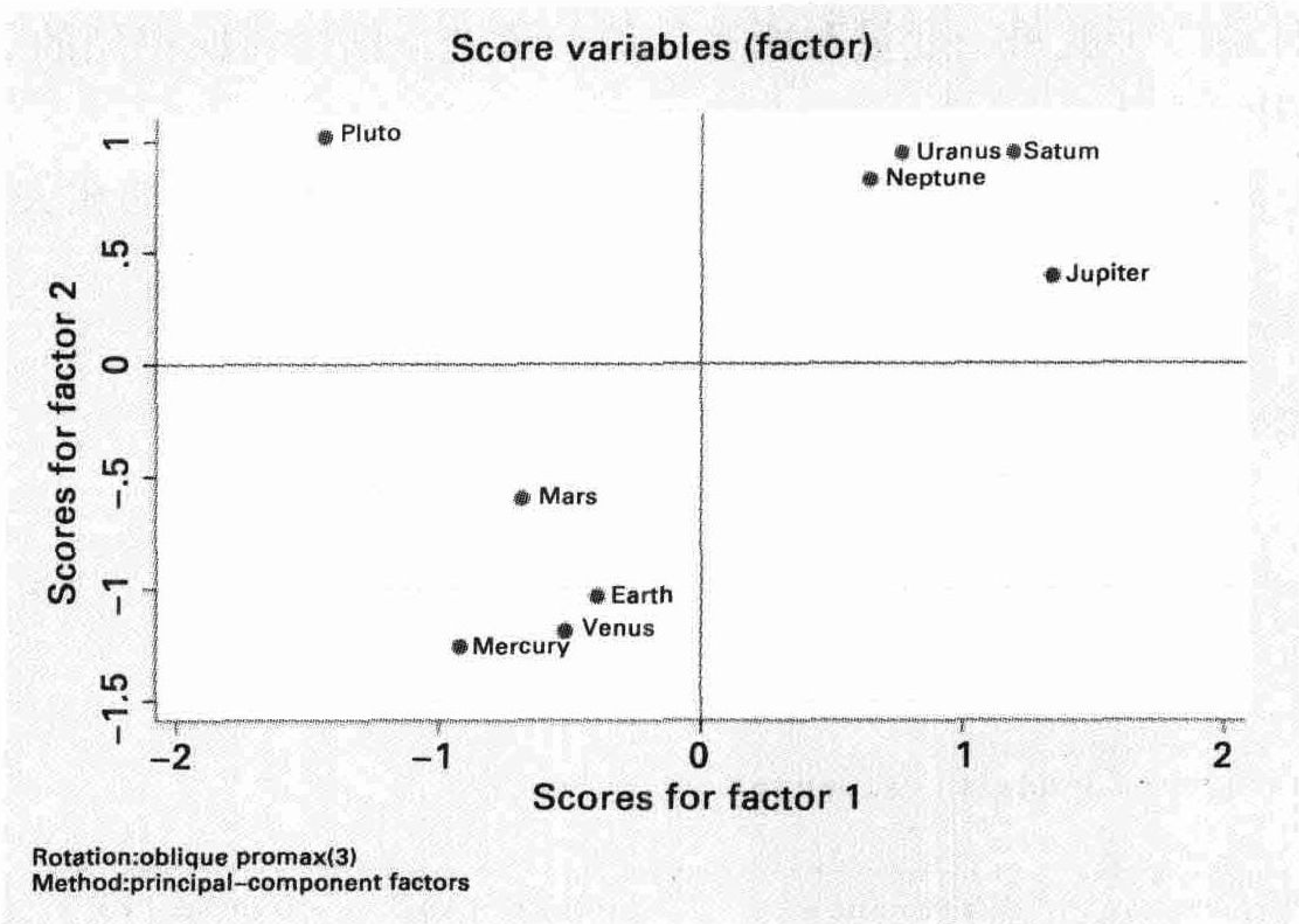


图 12.3

内层的那些岩质行星(比如,水星,在因子 1"遥远/低密度"上得分低;在因子 2"大规模/多卫星"上得分也低)聚集在图中的左下角。外层的气体巨星则具有相反的特征、并聚集在图中的右上角。冥王星(Pluto)在行星中是独一无二的,物理上类似于一些外层体系的卫星,在"遥远/低密度"维度上得分很高,但在"大规模/多卫星"维度上得分很低。我们的因子分析因此将冥王星看作与任一行星主群体都不一致的不同种类物体。由于认识到冥王星其实是个例外,国际天文学联盟 2006 年投票决定重新将它划归为诸多已知的"侏儒行星"之一,使得我们只有八个真正的行星。

如果采用最大方差而不是 promax 旋转的话,我们将得到相互独立的因子分:

```
. quietly factor rings logdsun - logdense, pcf
. quietly rotate
. quietly predict varimax1 varimax2
. correlate varimax1 varimax2
(obs =9)

             |  varimax1  varimax2
-------------+--------------------
    varimax1 |    1.0000
    varimax2 |    0.0000    1.0000
```

一旦由 **predict** 创建得到,因子分就能像任何其他 Stata 变量那样对其进行列出、计算相关、画图等操作。因子分在社会和行为科学中常常用来将许多测验或问卷项目合并成复合变量或指数。未旋转情况下采用主成分得到的因子分常用于分析来自气候学和遥感等自然科学领域的大型数据集。在这些应用中,主成分被称作"经验正交函数"(empirical orthogonal functions)。第一个经验正交函数即 EOF1 等于第一个未旋转的主成分因子分。第二个经验正交函数即 EOF2 则是第二个未旋转的主成分因子分,如此等等。

主因子法

主因子法(principal factoring)根据一个修正的相关矩阵提取主成分,这一矩

阵中的主对角线由公因子方差(communality)估计构成,而不是由 1 构成。**factor** 的两种选项 **pf** 和 **ipf** 都执行主因子法。它们在如何估计公因子方差上存在差别:

pf　　公因子方差估计值等于就每一变量对所有其他变量进行回归时所得到的 R^2。

ipf　　公因子方差的迭代估计。

尽管主成分分析集中于对变量的方差进行解释,但主因子法则是对变量之间的相关进行解释。

我们对行星数据采用按迭代公因子方差估计的主因子法(**ipf**):

```
. factor rings logdsun - logdense, ipf
(obs =9)

Factor analysis/correlation                        Number of obs    =        9
    Method: iterated principal factors             Retained factors =        5
    Rotation: (unrotated)                          Number of params =       15

    Beware: solution is a Heywood case
          (i.e., invalid or boundary values of uniqueness)

    --------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+------------------------------------------------------------
        Factor1  |      4.59663      3.46817            0.7903       0.7903
        Factor2  |      1.12846      1.05107            0.1940       0.9843
        Factor3  |      0.07739      0.06438            0.0133       0.9976
        Factor4  |      0.01301      0.01176            0.0022       0.9998
        Factor5  |      0.00125      0.00137            0.0002       1.0000
        Factor6  |     -0.00012                        -0.0000       1.0000
    --------------------------------------------------------------------------
    LR test: independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 = 0.0000

Factor loadings (pattern matrix) and unique variances

    -------------------------------------------------------------------------------
        Variable |  Factor1   Factor2   Factor3   Factor4   Factor5 |   Uniqueness
    -------------+--------------------------------------------------+--------------
           rings |   0.9760    0.0665    0.1137   -0.0206   -0.0223 |     0.0292
         logdsun |   0.6571   -0.6705    0.1411    0.0447    0.0082 |     0.0966
          lograd |   0.9267    0.3700   -0.0450    0.0486    0.0166 |    -0.0004
        logmoons |   0.9674   -0.0107    0.0078   -0.0859    0.0160 |     0.0564
         logmass |   0.8378    0.5458    0.0056    0.0282   -0.0071 |    -0.0007
        logdense |  -0.8460    0.4894    0.2059   -0.0061    0.0100 |     0.0022
    -------------------------------------------------------------------------------
```

本例中,Stata 给出了一个不妙的警告"Beware: solution is a Heywood case"。点击突出显示的 Heywood case 警告可得到对问题的解释,在这里,该问题反映出我们的样本量异常的小($n=9$)。出于简洁性的考虑,我们将继续这一分析,但是在实际研究时,这就需要重新考虑。

只有前两个因子具有大于 1 的特征值。采用 **pcf** 或 **pf** 方法,我们可以简单地忽略那些次要的因子。但是,在使用 **ipf** 时,我们必须决定要保留多少个因子,然后重复分析以准确地寻找那些因子。这里,我们将保留两个因子:

在 **ipf** 因子分析后,我们可以和以前一样通过 **rotate** 和 **predict** 创建复合变量。由于出现了 Heywood 情形的问题,这里的 **ipf** 因子分比我们之前 **pcf** 的结果更不合理。作为一种研究策略,使用不同的方法常常有助于重复因子分析,通过比较这些结果以得到稳定的结论。

```
. factor rings logdsun - logdense, ipf factor(2)
(obs =9)
```

```
Factor analysis/correlation                          Number of obs    =         9
    Method: iterated principal factors               Retained factors =         2
    Rotation: (unrotated)                            Number of params =        11

    Beware: solution is a Heywood case
            (i.e., invalid or boundary values of uniqueness)

    ------------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+----------------------------------------------------------------
        Factor1  |      4.57495      3.47412            0.8061       0.8061
        Factor2  |      1.10083      1.07631            0.1940       1.0000
        Factor3  |      0.02452      0.02013            0.0043       1.0043
        Factor4  |      0.00439      0.00795            0.0008       1.0051
        Factor5  |     -0.00356      0.02182           -0.0006       1.0045
        Factor6  |     -0.02537                        -0.0045       1.0000
    ------------------------------------------------------------------------------
    LR test: independent vs. saturated: chi2(15) =  100.49 Prob>chi2 = 0.0000

Factor loadings (pattern matrix) and unique variances

    -----------------------------------------------
        Variable |  Factor1   Factor2 |   Uniqueness
    -------------+--------------------+------------
           rings |   0.9747    0.0537 |     0.0470
         logdsun |   0.6533   -0.6731 |     0.1202
          lograd |   0.9282    0.3605 |     0.0086
        logmoons |   0.9685   -0.0228 |     0.0614
         logmass |   0.8430    0.5462 |    -0.0089
        logdense |  -0.8294    0.4649 |     0.0960
    -----------------------------------------------
```

最大似然法

和 Stata 的其他 **factor** 选项不同，最大似然因子法提供了正规的假设检验，该检验有助于确定合适的因子数目。为了得到适合于行星数据的一个单一的最大似然因子，键入：

. factor *fings logdsun* - *logdense*, ml nolog factor (1)

(obs =9)

```
Factor analysis/correlation                          Number of obs    =         9
    Method: maximum likelihood                       Retained factors =         1
    Rotation: (unrotated)                            Number of params =         6
                                                     Schwarz's BIC    =   97.8244
    Log likelihood = -42.32054                       (Akaike's) AIC   =   96.6411

    ------------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+----------------------------------------------------------------
        Factor1  |      4.47258                         1.0000       1.0000
    ------------------------------------------------------------------------------
    LR test: independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 = 0.0000
    LR test:    1 factor vs. saturated:  chi2(9)  =   51.73 prob>chi2 = 0.0000

Factor loadings (pattern matrix) and unique variances

    ---------------------------------------
        Variable |  Factor1 |   Uniqueness
    -------------+----------+--------------
           rings |   0.9873 |      0.0254
         logdsun |   0.5922 |      0.6493
          lograd |   0.9365 |      0.1229
        logmoons |   0.9589 |      0.0805
         logmass |   0.8692 |      0.2445
        logdense |  -0.7715 |      0.4049
    ---------------------------------------
```

ml 输出结果包括两个卡方检验：

似然比检验:独立模型对饱和模型(LR test: independence vs. saturated)

这检验一个无因子(独立)模型对观测相关矩阵的拟合是否显著地比一个饱和或完美拟合模型更差。较低的概率值(这里为0.000 0,意味着 $P < 0.00005$)表明无因子模型过于简单。

似然比检验:单因子模型对饱和模型(LR test: 1 factor vs. saturated)

这检验当前的单因子模型拟合得是否显著地比饱和模型更差。这里较低的 P 值表明一个因子也过于简单。

也许一个双因子模型会更好一些:

```
. factor rings logdsun - logdense, ml nolog factor(2)
(obs =9)

Factor analysis/correlation                    Number of obs    =        9
    Method: maximum likelihood                 Retained factors =        2
    Rotation: (unrotated)                      Number of params =       11
                                               Schwarz's BIC    =  36.6881
    Log likelihood = -6.259338                 (Akaike's) AIC   =  34.5187

    Beware: solution is a Heywood case
          (i.e., invalid or boundary values of uniqueness)

    ---------------------------------------------------------------------------
         Factor  |   Eigenvalue   Difference        Proportion   Cumulative
    -------------+-------------------------------------------------------------
        Factor1  |      3.64200      1.67115            0.6489       0.6489
        Factor2  |      1.97085                         0.3511       1.0000
    ---------------------------------------------------------------------------
    LR test: independent vs. saturated:  chi2(15) =  100.49 Prob>chi2 =  0.0000
    LR test:   2 factors vs. saturated:  chi2(4)  =    6.72 Prob>chi2 =  0.1513
    (tests formally not valid because a Heywood case was encountered)

Factor loadings (pattern matrix) and unique variances

    ------------------------------------------------
        Variable |  Factor1   Factor2 |   Uniqueness
    -------------+--------------------+-------------
           rings |   0.8655   -0.4154 |     0.0783
         logdsun |   0.2092   -0.8559 |     0.2236
          lograd |   0.9844   -0.1753 |     0.0003
        logmoons |   0.8156   -0.4998 |     0.0850
         logmass |   0.9997    0.0264 |     0.0000
        logdense |  -0.4643    0.8857 |     0.0000
    ------------------------------------------------
```

我们现在得到了以下结果:

似然比检验:独立模型对饱和模型(LR test: independence vs. saturated)

第一个检验没有变化;无因子模型过于简单。

似然比检验:双因子模型对饱和模型(LR test: 2 factors vs. saturated)

双因子模型并不显著地比完美拟合模型更差($P = 0.1513$)。

这些检验表明两个因子可以提供一个恰当的模型。

执行最大似然因子分析的计算程序常常产生 Heywood 解,即得出了负的方差或零独特性(zero uniqueness)等不切实际的结果。当出现这一现象时(正如我们的两因子 **ml** 例子中出现的那样),卡方检验缺乏正规合理性。但仅从描述来看,该检验仍能提供恰当的因子数目的非正规指引。

聚类分析—1

聚类分析(cluster analysis)包含多种将案例划分成不同组(groups)或类(clusters)的方法,它们都基于观测案例在许多变量上的相异性

(dissimilarities)。通常只是用它来做建立经验性类型的探索,而不是用它来检验事先所定的假设。实际上,对于普通的聚类方法而言,几乎没有什么指导假设检验的正规理论。分析中每一步可用选择的数量都多得惊人,而它们又很有可能导致许多不同的结果。本节只是提供了进行聚类分析的一个起点。我们先回顾一些基本思路,再通过一个简单的例子加以示范。在下一节中,我们将考虑一个略微更大一些的例子。Stata 的《多元统计参考手册》介绍并详细说明了全部可用的选择。Everitt 等(2001)更详细地讨论了这一主题,还包括许多聚类分析方法之间的有用比较。

聚类方法分成两个宽泛的类别:划分(partition)方法和层次(hierarchical)方法。划分方法将观测案例划分到一系列事先设定的不重合的分组中去。我们有两条途径做到这点:

cluster kmeans　　*K* 个平均数(mean)的聚类分析。

用于设定将要创建的聚类的数目(*K*)。Stata 然后通过迭代过程将观测案例分配到具有最接近的平均数的组从而找出这些聚类。

cluster kmedians　　*K* 个中位数(median)的聚类分析。

类似于 **Kmeans** 方法,但是采用中位数作为聚类标准。

划分聚类法在计算上往往比层次聚类法更简单且速度更快。但是对于探索性工作而言,事先必须指定聚类的精确数目的要求却又是一个缺点。

层次聚类法涉及使小群体逐渐融合形成大群体的一个过程。Stata 在层次聚类分析中采用一种聚集方式(agglomerative approach):它从视每一个观测案例为独立的"组"开始。最接近的两个组被合并,这一过程会不断进行,直到一个设定的停止点,或者是将全部观测案例归属于一个组。一种被称作系统树图或树状图的图形能将层次聚类结果可视化。有好几种联结方法(linkage method),它设定在包含多于一条观测案例的组之间应当进行比较的内容:

cluster singlelinkage　　最短联结法(single linkage)聚类分析。

将两个组之间最接近的一对观测案例之间的相异性(dissimilarity)作为两个组之间的相异性来加以计算。尽管简单,但是这一方法对特异值或测量错误的耐抗性(resistance)较差。观测案例是一次性聚类,往往形成非平衡的、不断加大的组。在这些组中,成员很少具有共性,但是又通过中间观测案例连结起来,这种问题被称作链接(chaining)问题。

cluster completelinkage　　最长联结法(complete linkage)聚类分析。

使用两组之间距离最远的一对观测案例作为代表。该方法对特异值没有最短联结法那样敏感,但具有相反的倾向,即容易将许多案例聚集成空间紧密的群。

cluster averagelinkage　　平均联结法(average linkage)聚类分析。

使用两个组之间观测案例的平均相异性,产生的属性居于最短联结法和最长联结法之间。模拟研究报告表明,这一方法在许多情况下都表现很好,并且合理地稳健(见 Everitt 等,2001,以及他们所引用的文献)。这种方法常用于考古学中。

cluster waveragelinkage　　加权平均联结法(weighted-average linkage)聚类分析。

cluster medianlinkage 中位数联结法(median linkage)聚类分析。

加权平均联结法和中位数联结法分别是平均联结法和重心联结法的变种。在这两种情形中,差异在于不等规模的组在合并时是如何处理的。对于平均联结法和重心联结法来说,每一组元素的数量被分解到计算中,并对更大的组相应地赋予更大的影响(因为每条观测案例权数相同)。对于加权平均联结法和中位数联结法而言,不管每组中有多少观测案例,两个组都被赋予相同的权数。同重心联结法一样,中位数联结法也很容易受到逆转的影响。

cluster centroidlinkage 重心联结法(centroid average linkage)聚类分析。

重心法合并那些平均数最为接近的组(与基于两组元素之间平均距离的平均联结法不同)。这一方法容易发生逆转(reversals),即某次聚合的点比前面的聚合的相异性水平更低。逆转是聚类结构不稳定的迹象,它难以解释,并且不能用 **cluster tree** 画出来。

cluster wardslinkage Ward 的联结法(Ward's linkage)聚类分析。

合并能使误差平方和(error sum of squares)增加最少的两个组。尽管可以适当地处理那些多元正态和相似规模的组,但是在聚类具有不相等的观测案例数时表现较差。

所有的聚类方法都从相异性(或相似性)的某一定义入手。相异性指标反映了两个观测案例在设定的一套变量上的差异或距离。总而言之,这种指标在两个相同的观测案例上测量的相异性为 0,而两个最大差别的观测案例具有的相异性为 1。相似性指标正好相反,因此相同的案例的相似性为 1。Stata 的 **cluster** 选项提供了相异性或相似性测量的许多选择。出于计算目的,Stata 内在地将相似性转换成相异性:

相异性 = 1 - 相似性

默认的相异性指标是欧氏距离(Euclidean distance),即选项 **L2**(或 **Euclidean**)。它将观测案例 i 和 j 之间的距离定义为:

$$\left\{\sum_k (x_{ki} - x_{kj})^2\right\}^{1/2}$$

其中,x_{ki} 是观测案例 i 在变量 x_k 上的取值,x_{kj} 是观测案例 j 在变量 x_k 上的取值,合计号针对所有被考虑的 x 变量进行。其他基于连续变量测量观测案例之间的相似(异)性的可用选择还包括欧氏距离的平方(**L2 squared**),即

$$\sum_k (x_{ki} - x_{kj})^2$$

以及绝对值距离(**L1**)、最大值距离(**Linfinity**)和相关系数相似性测量(**correlation**)。基于二分变量(binary variables)的相异性或相似性的选择包括简单匹配(**matching**)、Jaccard 二分类相似系数(**Jaccard**)以及许多其他选择。请键入 **help meansure_option** 查看清单和解释。

在本章的前面,*planets.dta* 中变量的主成分分析鉴别出三种类型的行星:内层的坚硬行星、外层的气体巨星和自成一类的冥王星(Pluto)。聚类分析提供了回答行星"类型"问题的替代途径。因为诸如卫星数量(*moons*)和以千克测量的质量(*mass*)等这些变量都是以不可比的单位进行的测量,具有极为不同的方差,因此我们应当以某种方式进行标准化以避免结果受到具有最大方差的项目的影响。一个常用的但不是自动的选择就是做平均数为零和标准差为 1 的标准化。这可以通过 **egen** 命令来实现(出于

和前面讨论中相同的理由,使用对数形式的变量)。**summarize** 确认新的 *z* 变量具有(近似的)均值为零和标准差为 1。

```
. egen zrings = std(rings)
. egen zlogdsun = std (logdsun)
. egen zlograd = std (lograd)
. egen zlogmoon = std (logmoons)
. egen zlogmass = std (logmass)
. egen zlogdens = std (logdense)
. summ zrings - zlogdens
```

Variable	Obs	Mean	Std. Dev.	Min	Max
zrings	9	-1.99e-08	1	-.8432741	1.054093
zlogdsun	9	-1.16e-08	1	-1.393821	1.288216
zlograd	9	-3.31e-09	1	-1.3471	1.372751
zlogmoon	9	0	1	-1.207296	1.175849
zlogmass	9	-4.14e-09	1	-1.74466	1.365167
zlogdens	9	-1.32e-08	1	-1.453143	1.128901

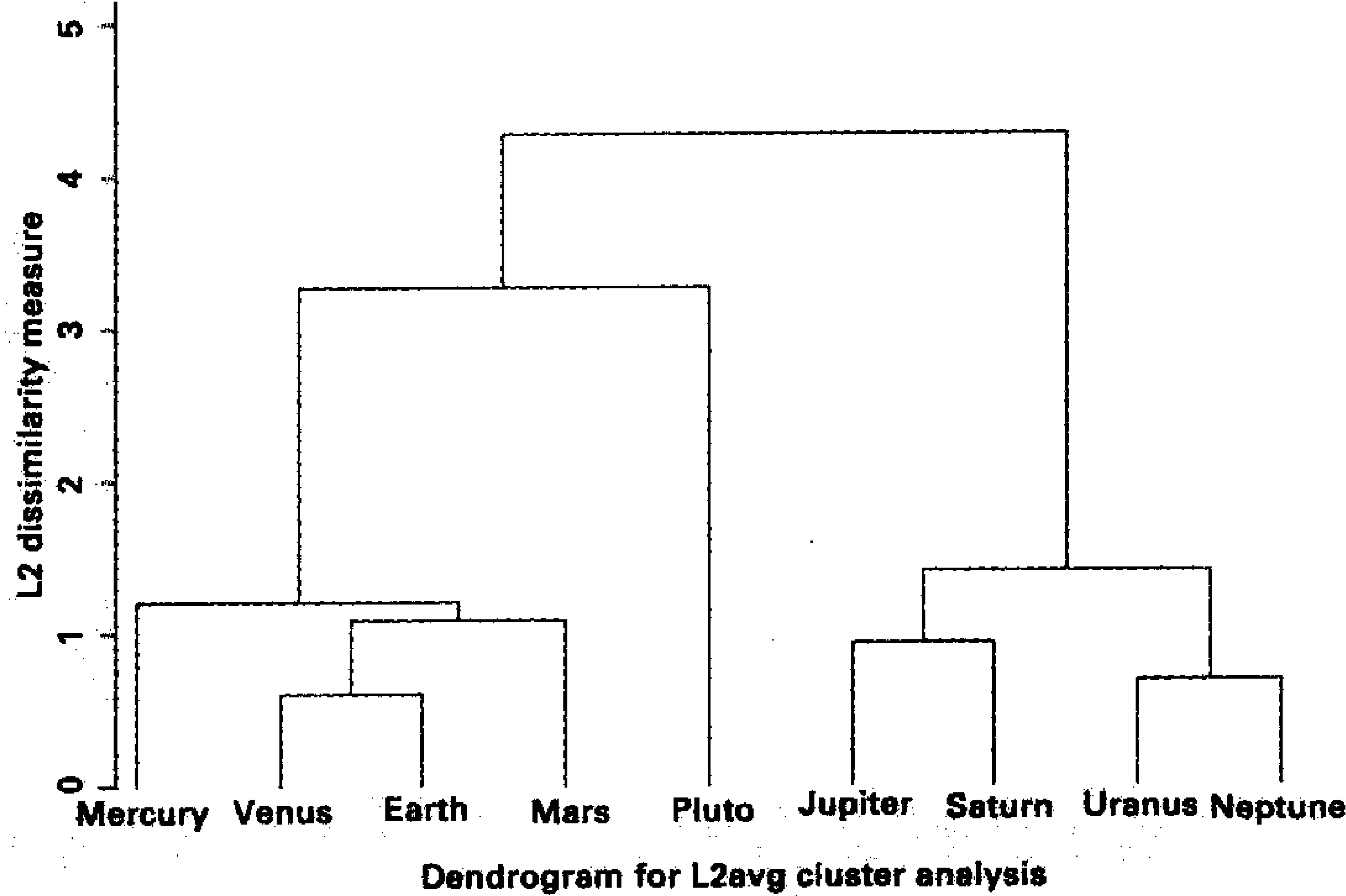
主成分分析表明的"三种类型"的结论是稳健的,这也能通过聚类分析来得到。比如,我们可以使用欧氏距离(L2)作为相异性测量、并采用平均联结法进行层次聚类分析。选项 **name(*L2avg*)** 赋予得自这一特定分析结果的一个变量名,因此我们能够在随后的命令中引用它们。当我们需要尝试大量的聚类分析并对其结果进行比较时,能对结果进行命名的特点就提供了方便。

```
. cluster averagelinkage zrings zlogdsun zlograd zlogmoon zlogmass
    zlogdens, L2 name (L2avg)
```

似乎什么都没有发生,尽管我们可能注意到我们的数据集现在包含三个具有基于 *L2avg* 的名称的新变量。这些新的 *L2avg** 变量并不是我们所直接关注的,但是可以用 **cluster dendrogram** 命令来画出聚类分析树状图或系统树图,将最近的层次聚类分析结果可视化(图 12.4)。这里的 **label(*planet*)** 选项使得行星名称(即 *planet* 的取值)在下面的树状图中作为标签显示。

```
. cluster dendrogram, label (planet) ylabel (0(1)5)
```

图 12.4



像图 12.4 这样的系统树图提供了层次聚类分析的主要解释工具。我们能够从底部追溯每条观测案例作为自身的聚类到顶部的所有观测案例聚合成一个聚类的聚集过程。金星(Venus)和地球(Earth)以及天王星(Uranus)和海王星(Neptune)都是最少差异或最为相似的对(pairs)。它们首先聚合,在高度(即相异性,dissimilarity)低于 1 处形成了最早的两个多观测案例聚类。木星(Jupiter)和土星(Saturn),然后是金星—地球和火星,然后是金星—地球—火星和水星,最后是木星—土星和天王星—海王星接二连三地都在相异性为 1 左右的位置聚合。在这点处,图 12.3 已经建议了与主成分分析同样的三个组:内层坚硬的行星、气体巨星和冥王星。这三个聚类在更高相异性处(大于 3)还仍然保持稳定,直到冥王星和内层坚硬的行星聚合在一起。在相异性接近 4 的水平时,最后的两个聚类聚合了。

那么,到底有多少个行星类型呢? 正如图 12.4 所说明的,答案是"要看情况而定"。也就是说,看我们想接受每一类型内多大的相异性。图中上部的三聚类阶段和二聚类阶段之间的长长垂线表明我们有三种相当不同的类型。我们也可以将此减少到两类,仅需通过聚合与其同组中的其他行星极不相似的观测案例(冥王星)。我们还可以扩展到五种类型,只需要画出这些行星组(比如,水星—火星与地球—金星)之间的差别,但是按太阳系的标准它们之间的差别并不太大。因此,这个系统树图提供了一个三类型方案的例子。

命令 **cluster generate** 创建一个新变量,用以标示每一观测案例所属类型或组。在本例中,**group(3)**要求的是三个组。**name(L2avg)**选项设定了我们命名为 *L2avg* 的特定结果。当我们这一段操作中包含了多个聚类分析时候,这一选项就非常有用了。

```
. cluster generate plantype = groups(3), name (L2avg)
. label variable plantype "Planet type"
. list planet plantype
```

```
     +--------------------+
     |  planet   plantype |
     |--------------------|
  1. | Mercury          1 |
  2. |   Venus          1 |
  3. |   Earth          1 |
  4. |    Mars          1 |
  5. | Jupiter          3 |
     |--------------------|
  6. |  Saturn          3 |
  7. |  Uranus          3 |
  8. | Neptune          3 |
  9. |   Pluto          2 |
     +--------------------+
```

内层坚硬的行星已被编码成 plantype = 1,气体巨星被编码成 plantype = 3,而比其他行星更像外层系统卫星的冥王星被单独编码成 plantype = 2。组别分配为 1、2 和 3 是按照系统树图(图 12.4)中最终聚类从左到右的排序。一旦数据被保存,我们的新类型在随后的分析中就可以像任何其他分类变量那样来加以使用。

这些行星数据具有一种很强的自然分组模式,这就是为什么诸如聚类分析和主成分分析这些不同的技术都得到类似结果的原因所在。我们还可以对这个例子选择其他的相异性测量和联结法,仍然会得到极为相似的结果。但是,用复杂或缺乏模式的数据时,由于所用方法的细微差别便常常导致极为不同的结果。由一种方法得到的聚类可能并不能被其他方法重复,甚至在分析中一些细微的设置不同也会影响最终结果。

聚类分析—2

发现一个简单、稳健的描述九颗行星的类型较为简单。另一个更具挑战性的例子将考察 nations.dta 中的跨国数据。该数据集包含了生活条件变量,它们提供了将这些国家区分为不同类别的基础信息。

```
Contains data from C:\data\nations.dta
  obs:           109                          Data on 109 nations, ca. 1985
 vars:            15                          2 Jan 2008 13:31
 size:         4,578 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
country         str8   %9s                    Country
pop             float  %9.0g                  1985 population in millions
birth           byte   %8.0g                  Crude birth rate/1000 people
death           byte   %8.0g                  Crude death rate/1000 people
chldmort        byte   %8.0g                  Child (1-4 yr) mortality 1985
infmort         int    %8.0g                  Infant (<1 yr) mortality 1985
life            byte   %8.0g                  Life expectancy at birth 1985
food            int    %8.0g                  Per capita daily calories 1985
energy          int    %8.0g                  Per cap energy consumed, kg oil
gnpcap          int    %8.0g                  Per capita GNP 1985
gnpgro          float  %9.0g                  Annual GNP growth % 65-85
urban           byte   %8.0g                  % population urban 1985

school1         int    %8.0g                  primary enrollment % age-group
school2         int    %8.0g                  Secondary enroll % age-group
school3         byte   %8.0g                  Higher ed. enroll % age-group
-------------------------------------------------------------------------------
Sorted by:
```

在第 8 章中,我们看到非线性转换(取对数或平方根)有助于将分布加以正态化以及将一些变量之间的关系加以线性化。非线性转换的类似思路也能应用于聚类分析,不过为了使我们的例子简单,我们将不在这里细究。但是,以某种形式将变量做标准化的线性转换仍然是重要的。否则,人均国内产值变量 *gnpcap* 的取值范围从 100 美元到 19 000美元(标准差为 4 400 美元),这将淹没像取值范围从 40 年到 78 年(标准差为 11 年)的预期寿命 life 等其他变量。在上一节,我们通过减去每一变量的平均数然后除以它们的标准差来标准化行星数据,因此作为结果的所有 *z* 分的标准差全都为 1。在本节中,我们将采用一种不同的方法,即全距标准化(range standardization),这种方法对聚类分析也能起很好作用。

全距标准化对每个变量除以自己的全距。Stata 中没有相应的直接命令,但是我们能很容易地临时准备一个。**summarize, detail** 命令可以计算单变量的统计量,然后我们将这些结果作为宏(将第 14 章中介绍)暂存于内存中。名为 r(max) 的宏暂存变量的最大值,名为 r(min) 的宏暂存其最小值。于是,为了创建新变量 *rpop*,即变量 *pop*(人口)的全距标准化值,键入命令:

```
. quietly summ pop, detail
. generate rpop = pop/(r(max) - r(min))
. label variable rpop "Range-standardized population"
```

用类似的命令再创建其他生活条件变量的全距标准化值:

```
. quietly summ birth, detail
. generate rbirth = birth/(r(max) - r(min))
. label variable rbirth "Range - standardized bith rate"
```

```
. quietly summ infmort, detail
. generate rinf = infmort/(r(max) - r(min))
. label variable rinf "Range-standardized infant mortality"
```

如此等等,就定义了以下所列的 8 个新变量。这些全距标准化变量都具有等于 1 的全距。

```
. describe rpop - rschool2

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
rpop            float  %9.0g                  Range-standardized population
rbirth          float  %9.0g                  Range-standardized bith rate
rinf            float  %9.0g                  Range-standardized infant
                                                mortality
rlife           float  %9.0g                  Range-standardized life
                                                expectancy
rfood           float  %9.0g                  Range-standardized food per
                                                capita
renergy         float  %9.0g                  Range-standardized energy per
                                                capita

rgnpcap         float  %9.0g                  Range-standardized GNP per
                                                capita
rschool2        float  %9.0g                  Range-standardized secondary
                                                school %
```

```
. summarize rpop - rschool2

    Variable |     Obs        Mean    Std. Dev.       Min        Max
-------------+-------------------------------------------------------
        rpop |     109    .0374493    .1206474    .0009622   1.000962
      rbirth |     109    .7452043    .3098672    .2272727   1.227273
        rinf |     109    .4051354    .2913825     .035503   1.035503
       rlife |     109    1.621922     .291343    1.052632   2.052632
       rfood |     108    1.230213    .2644839    .7793776   1.779378
-------------+-------------------------------------------------------
     renergy |     107     .159786    .2137914    .0018464   1.001846
     rgnpcap |     109    .1666459    .2319276    .0057411   1.005741
    rschool2 |     104    .4574849    .2899882    .0196078   1.019608
```

在关注变量已经被标准化后,我们就可以继续进行聚类分析。尽管我们将 100 多个国家区分成"类型",但是我们没有理由假定每个类型将包含同样多的国家。和其他方法一样,平均联结法(就是我们在行星例子中所采用的)赋予每个观测案例同样的权数。随着聚集的进行,使得更大的聚类便更有影响。但是,加权平均法和中位数联结法是赋予每一聚类同等权数,而不管它包含多少个观测案例。因此,此类方法往往对探测不等规模的聚类具有更好的效果。如同重心联结一样,中位数联结也容易受到逆转的影响(在这些数据中将会发生),因此下面的例子采用加权平均联结法。绝对值距离(**L1**)提供了我们相异性的测量。

```
. cluster waveragelinkage rpop - rschool2 , L1 name (L1wav)
```

完整的因子分析提供的树状图被证明大得难以处理:

```
. cluster dendrogram
too many leaves; consider using the cutvalue( ) or cutnumber ( ) options
r(198);
```

根据错误信息提示,在起初出现少数聚合之后,图 12.5 就采用 **cutnumber(100)** 选项形成了从只有 100 个组开始的系统树图。

```
. cluster dendrogram, ylabel (0(.5)3) cutnumber(100)
```

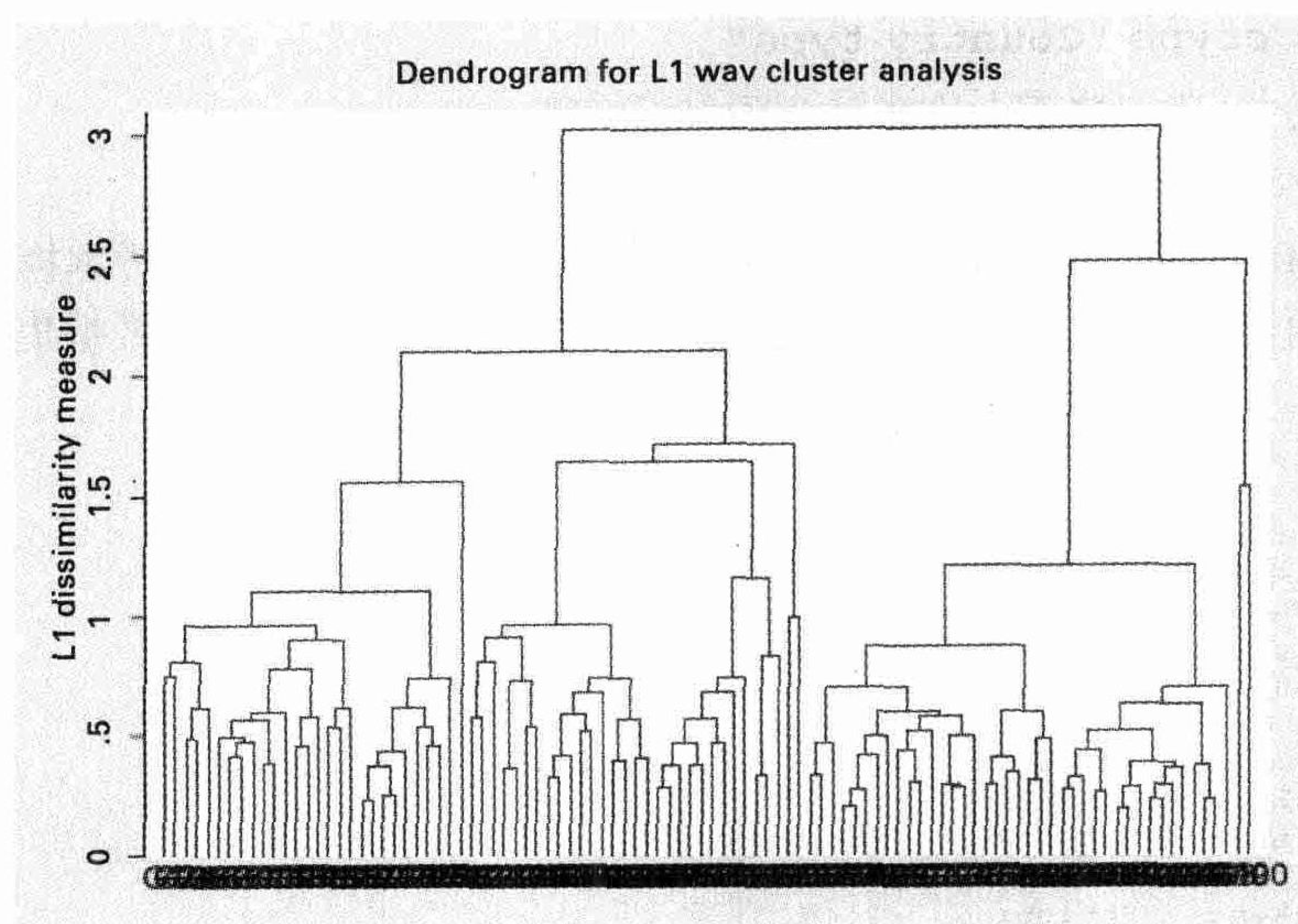


图 12.5

图 12.5 中底部的标签根本没办法读，但是我们能够追溯这一聚类过程的一般流程。大部分聚合都发生在相异性低于 1 水平。最右边的两个国家很不寻常；它们直到大约 1.5水平才聚合，然后形成一个完全不同于所有其他组的稳定的两国之组。这是四个相异性仍然大于 2 的聚类中的一个。这四个最终聚类（从左到右）中的第一和第二个表现出异质性，它们经过大量略微不同于大多数子群的连续聚合而形成。相比而言，第三个聚类显得更具同质性。它合并了在相异性低于 1 处聚合成两个子群的许多国家，然后略微高于 1 处聚合成一个组。

图 12.6 给出了这一分析的另一种视角，这次使用 **cutvalue(1)** 选项，即只显示那些相异性高于 1 的聚类。**xlabel(, angle(vertical))** 选项，这里实际上并不需要，要求垂直而不是水平地显示底部标签（G1、G2 等）。

```
. cluster dendrogram, ylabel (0(.5)3) cutvalue(1)
  xlabel(, angle(vertical))
```

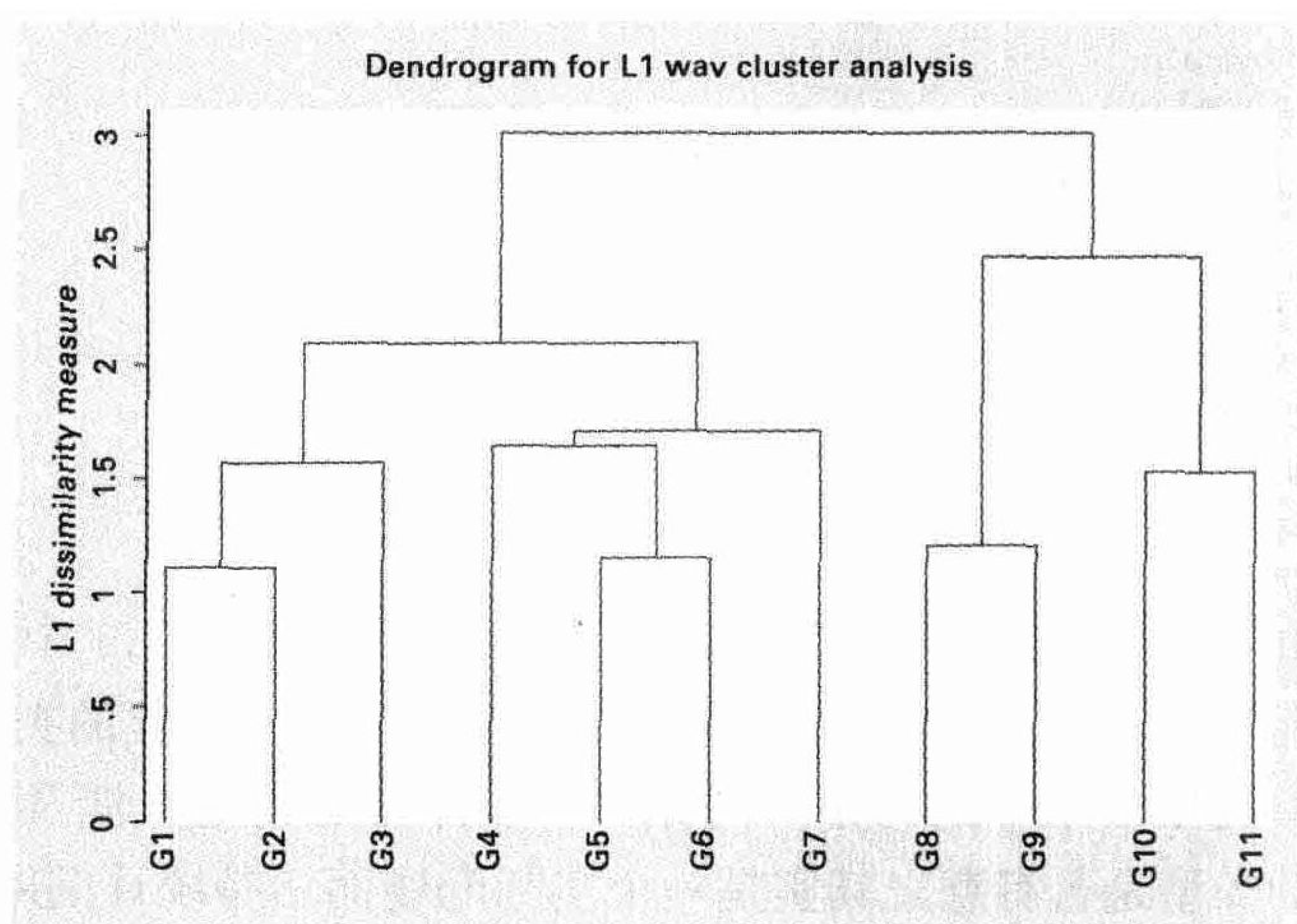


图 12.6

正如图 12.6 显示的那样，在相异性大于 1 处仍然有 11 个组。出于示范目的，我们将只考虑顶部相异性高于 2 的 4 个组。**cluster generate** 将根据上述我们称为 *L1 wav* 的聚类分析的最后四个组创建了一个分类变量。

```
. cluster generate ctype = groups (4), name (L1 wav)
```

```
. label variable ctype "Country type"
```

通过键入以下命令[16],我们接下来检查每个国家属于哪一组:

```
. by ctype: list country
```

具有相同信息的更简洁的列表呈现在下面。这一列表通过将 `nations.dta` 的数据拷贝和粘贴到数据编辑器中而得到,形成了一个以国家类型作为列的不同的、单一目的数据集。

```
     +-------------------------------------------+
     |   ctype1     ctype2     ctype3    ctype4  |
     |-------------------------------------------|
  1. |  Algeria   Argentin   Banglade     China  |
  2. |   Brazil   Australi      Benin     India  |
  3. |    Burma    Austria    Bolivia            |
  4. |    Chile    Belgium   Botswana            |
  5. | Colombia     Canada   BurkFaso            |
     |-------------------------------------------|
  6. | CostaRic    Denmark    Burundi            |
  7. |   DomRep    Finland   Cameroon            |
  8. |  Ecuador     France   CenAfrRe            |
  9. |    Egypt     Greece   ElSalvad            |
 10. | Indonesi   HongKong   Ethiopia            |
     |-------------------------------------------|
 11. |  Jamaica    Hungary      Ghana            |
 12. |   Jordan    Ireland   Guatemal            |
 13. | Malaysia     Israel     Guinea            |
 14. | Mauritiu      Italy      Haiti            |
 15. |   Mexico      Japan   Honduras            |
     |-------------------------------------------|
 16. |  Morocco     Kuwait   IvoryCoa            |
 17. |   Panama   Netherla      Kenya            |
 18. | Paraguay   NewZeala    Liberia            |
 19. |     Peru     Norway   Madagasc            |
 20. | Philippi     Poland     Malawi            |
     |-------------------------------------------|
 21. | SauArabi   Portugal   Mauritan            |
 22. | SriLanka    S_Korea   Mozambiq            |
 23. |    Syria   Singapor      Nepal            |
 24. | Thailand      Spain   Nicaragu            |
 25. |  Tunisia     Sweden      Niger            |
     |-------------------------------------------|
 26. |   Turkey   TrinToba    Nigeria            |
 27. |  Uruguay        U_K   Pakistan            |
 28. | Venezuel      U_S_A    PapuaNG            |
 29. |            UnArEmir     Rwanda            |
 30. |            W_German    Senegal            |
     |-------------------------------------------|
 31. |            Yugoslav   SierraLe            |
 32. |                        Somalia            |
 33. |                          Sudan            |
 34. |                       Tanzania            |
 35. |                           Togo            |
     |-------------------------------------------|
 36. |                         YemenAR           |
 37. |                       YemenPDR            |
 38. |                          Zaire            |
 39. |                         Zambia            |
 40. |                       Zimbabwe            |
     +-------------------------------------------+
```

图 12.5 中最右边所看到的两国聚类结果就是类型 4,有中国和印度。图 12.5 中宽泛的、同质的第三聚类即类型 3 涉及一大群主要在非洲的最穷的国家。相对多样化的类型 2 包括美国、欧洲和日本,是具有更高生活水平的国家。同样具有多样化的类型 1 涉及那些中等水平的国家。这种或其他的类型是否有意义其实是一个实际问题而不是统计问题,并且取决于这一类型的用途。在聚类分析的步骤中挑选不同的选项可能得到不同的结果。通过尝试不同的合理选择,我们就可以取得一种关于哪些结果最为稳定的理解。

[16]【译注:该命令需要先运行 **sort** *ctype* 对数据按照变量 *ctype* 进行排序后才能使用。或者,可直接改用 **by** *ctype*,**sort**:**list** *country* 命令或改用 **bysort** *ctype*:**list** *country* 也行。】

13 时间序列分析

Stata 的《时间序列参考手册》关于时间序列的功能就有 350 页。本章只提供一个简要的介绍,着手于两个基础性而又实用的分析工具:时间的作图和修匀。然后,我们将示范相关图、ARIMA 模型以及对稳态和白噪声的检验。更多的应用如周期图和灵活的 ARCH 家族模型将留给读者自己去探索。

关于时间序列的技术以及全套处理可以参见 Hamilton(1994)。其他参考文献还有 Box、Jenkins 和 Reinsel(1994),Chatfield(1996),Diggle(1990),Enders(1995),Johnston 和 DiNardo(1997),以及 Shumway(1988)。

时间序列的操作菜单有下列标题:

Statistics-Time series	时间序列
Statistics-Multivariate time series	多元时间序列
Statistics-Longitudinal /panel data	纵贯及面板数据
Graphics-Time series graphs	时间序列图形

命令示范

```
. ac y, lags(8) level(95) generate(newvar)
```

对变量 *y* 的自相关进行绘图,包含 95% 置信区间(默认),按照时滞 1 至 8。将自相关作为新变量 *newvar* 的前 8 个值来暂存。

```
. arch D.y, arch(1/3) ar(1) ma(1)
```

为 *y* 的一阶差分拟合一个 ARCH 模型(autoregressive conditional heteroskedasticity model,即自回归条件异方差模型),包括 ARCH 的一阶至三阶项,以及一阶的 AR 和 MA 扰动项。

```
. arima y, arima(3,1,2)
```

拟合一个简单的 ARIMA(3,1,2)模型。可能的选项包括几种不同估计方法、线性约束以及稳健方差估计。

```
. arima y, arima(3,1,2) sarima(1,0,1,12)
```

拟合一个 ARIMA 模型,内含一个按 12 个时期划分的季节乘数分量。

```
. arima D.y x1 L1.x1 x2, ar(1) ma(1 12)
```

将 *y* 的一阶差分对 *x1*、*x1* 的时滞 1(lag-1)的值以及 *x2* 做回归,其中包括AR(1)、MA(1)和 MA(12)扰动项。

```
. corrgram y, lags(8)
```

取得自相关、偏自相关,并且对时滞 1 ~8 做 *Q* 检验。

```
. dfuller y
```

进行 Dickey-Fuller 单位根的稳态检验。

```
. dwstat
```

在执行 **regress** 以后,计算 Durbin-Watson 统计量来检验一阶自相关。

```
. egen newvar = ma(y), nomiss t(7)
```

建立新变量 *newvar*,等于跨距 7 的 *y* 移动平均数,用较短、未对中的平均数取代起点值和终点值。

```
. generate date = mdy(month,day,year)
```

创建变量 *date*,其值为根据月(*month*)、日(*day*)、年(*year*)三个变量计算的自 1960 年 1 月 1 日以来的消逝天数。

```
. generate date = date(str_date, "mdy")
```

创建变量 *date*,其值为字符串变量 *str_date* 所转换的消逝天数。字符串变量 *str_date* 中包含月、日、年的日期信息,比如,"11/19/2001"、"4/18/98"或"June 12, 1948"。键入 **help dates** 咨询有关各种日期函数及其选项。

```
. generate newvar = L3.y
```

建立新变量 *newvar*,等于 *y* 的时滞 3(lag-3)的值。

```
. pac y, lags(8) yline(0) ciopts(bstyle(outline))
```

画出时滞 1 到 8 的带置信区间和残差方差的偏自相关图。图中加入 0 值水平线,显示置信区间为轮廓而不是阴影区域(默认)。

```
. pergram y, generate(newvar)
```

画出变量 *y* 的样本周期图(谱密度函数)并且创建变量 *newvar* 等于周期图的粗值。

```
. prais y x1 x2
```

将 *y* 对 *x1*,*x2* 做 Prais-Winsten 回归,以修正一阶自回归误差。此外,**prais y x1 x2, corc** 采用 Cochrane-Orcutt 转换进行估计。

```
. smooth 73 y, generate(newvar)
```

创建变量 *newvar*,等于跨距 7 的 y 移动中位数(running median),再按跨距 3 的移动中位数作修匀。复合校平器如"3RSSH"或"4253h,twice"也是可能的。键入 **help smooth** 或 **help tssmooth** 查询其他修匀方法及过滤器。

```
. tsset date, format(%d)
```

将数据集设置为时间序列。时间用变量 *date* 来表示,时期单位为天(daily)。对于"面板"(panel)数据,即一组不同的单位如城市有平行的时间序列,**tsset *city year*** 同时指定了面板变量和时间变量。本章的大多数命令都要求数据经过 **tsset** 处理。

. **tssmooth ma *newvar* = *y*, window(2 1 2)**

对 *y* 应用移动平均过滤器,创建变量 *newvar*。选项 **window(2 1 2)** 通过在每一修匀点的计算中用 2 个时滞值、当前观测和 2 个前导值求出跨距 5 的移动平均数。键入 **help tssmooth** 查询其他可用的过滤器,包括加权移动平均数、指数或双指数、Holt-Winters以及非线性等过滤器。

. **tssmooth nl *newvar* = *y*, smoother(4253h,twice)**

对 *y* 应用一个非线性修匀过滤器,创建 *newvar*。选项 **smoother(4253h,twice)** 迭代求出跨距为 4,2,5,3 的移动中位数,然后应用 Hanning 加权函数,然后再重复这个过程于残差。**tssmooth nl**,与其他 **tssmooth** 程序不同,它在缺失值周围不能工作。

. **wntestq *y*, lags(15)**

对白噪声进行 Box-Pierce 混合 *Q* 检验(**corrgram** 也可做此检验)。

. **xcorr *x* *y*, lags(8) xline(0)**

画出输入变量 *x* 和输出变量 *y* 之间时滞为 1-8 的交互相关图。**xcorr x y, table** 可以提供文本方式的输出,其中包括实际相关(如果包括 **generate(*newvar*)** 选项便将这些相关存为一个变量)。

修 匀

许多时间序列数据都展示忽上忽下的波动,以至很难辨别背后的模式。修匀(smoothing)这样的序列就是将数据分解为两部分,一部分为逐渐的变化,另一部分是"粗糙"部分,包含那些其余的迅速变化:

数据 = 修匀部分 + 粗糙部分

数据文件 *MILwater.dta* 包含了美国新罕布什尔州米尔福镇 1983 年前 7 个月的日常用水数据(Hamilton, 1985b)。

```
Contains data from MILwater.dta
  obs:           212                         Milford daily water use, 1/1/83
                                               - 7/31/83
 vars:             4                         27 Jul 2005 12:41
 size:         2 120  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
month           byte   %9.0g                  Month
day             byte   %9.0g                  Date
year            int    %9.0g                  Year
water           int    %9.0g                  Water use in 1000 gallons
-------------------------------------------------------------------------------
Sorted by:
```

在进行分析以前,我们需要将月日年信息转换为表示时间的一个数量指标。Stata 的 **mdy( )** 函数可以做这种工作,创建一个消逝天数变量(这里称为 *date*),表示自 1960 年 1 月 1 日以来的天数。

. **generate *date* = mdy(*month*,*day*,*year*)**

. **list in 1/5**

```
     +--------------------------------------+
     | month    day    year   water    date |
     |--------------------------------------|
  1. |     1      1    1983     520    8401 |
  2. |     1      2    1983     600    8402 |
  3. |     1      3    1983     610    8403 |
  4. |     1      4    1983     590    8404 |
  5. |     1      5    1983     620    8405 |
     +--------------------------------------+
```

作为参照日期的1960年1月1日是一个硬性规定的默认值。我们可以为 *date* 提供更好理解的格式化，再将我们的数据设置好以便后面的分析，所用的命令 **tsset**(意为时间序列设置，time series set)将 *date* 作为时间指标变量来识别，并且指定这一变量的显示格式为**% d**(其中 **d** 代表日常(daily)格式)。

```
. tsset date, format(%d)
        time variable:  date, 01jan1983 to 31jul1983

. list in 1/5

     +-------------------------------------------+
     | month   day   year   water         date |
     |-------------------------------------------|
  1. |     1     1   1983     520    01jan1983 |
  2. |     1     2   1983     600    02jan1983 |
  3. |     1     3   1983     610    03jan1983 |
  4. |     1     4   1983     590    04jan1983 |
  5. |     1     5   1983     620    05jan1983 |
     +-------------------------------------------+
```

新变量 *date* 的日期格式，如"05 jan 1983"比起原来的数值如"8405"(即自1960年1月1日以来的天数)可读性更强。如果需要，我们也可以用 %d 格式化得到其他格式，比如，"05 Jan 1983"或者"01/05/83"。Stata 提供很多种变量定义、显示格式、数据集格式等，这些对于时间序列分析都很重要。这些当中有许多涉及日期的输入、转换和显示。全部的日期函数描述请参见《数据管理参考手册》和《用户指南》，或者在 Stata 内键入 **help dates** 进行探索。

在 *water* 和 *date* 的两维标绘图中，*date* 值被加上日期标签，可以看出用水量按天的变化，同样也可以看出一到夏天用水量就处于上升趋势(图13.1)：

```
. graph twoway line water date, ylabel(300(100)900)
```

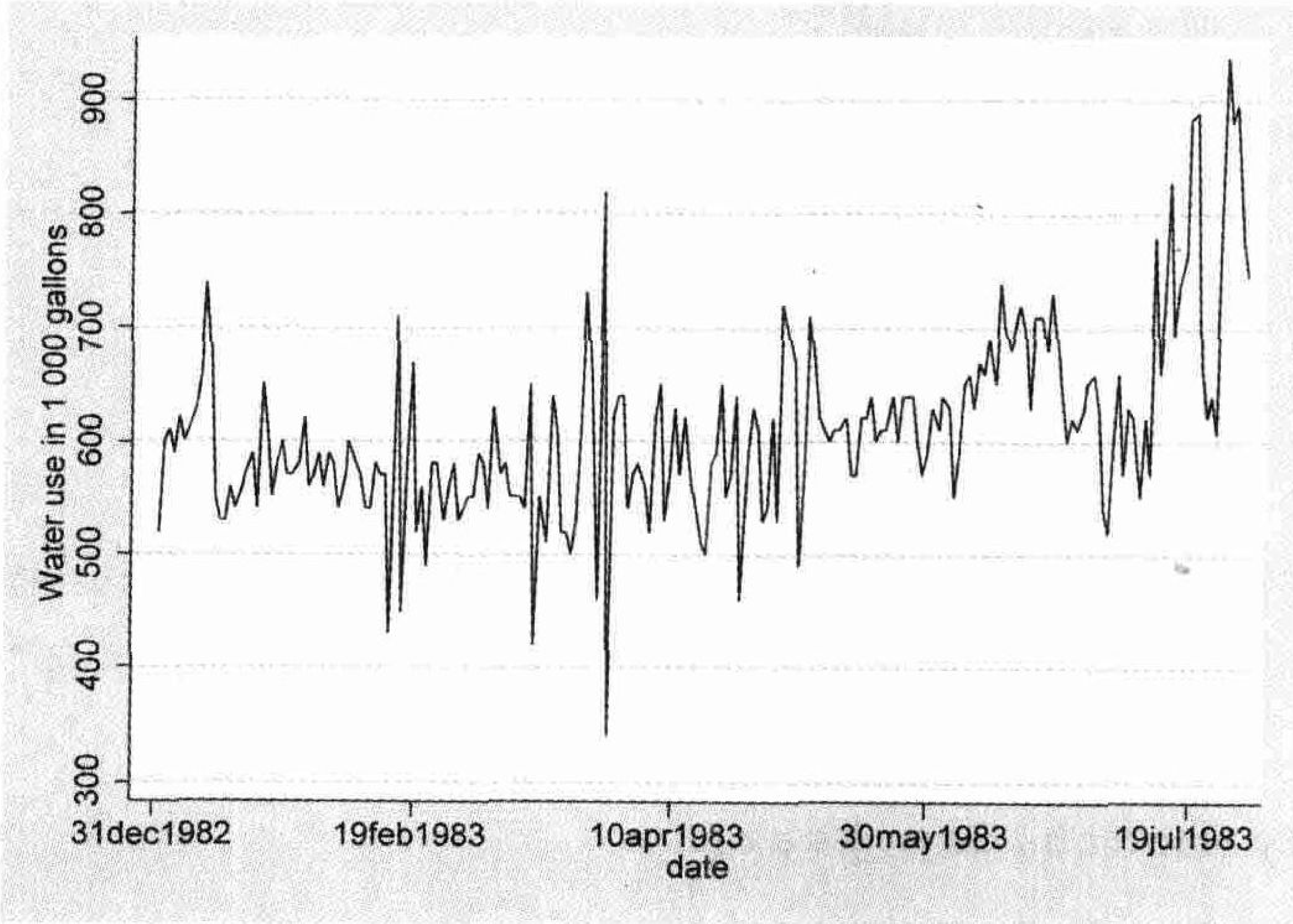


图13.1

视觉检查在时间序列分析中起重要的作用。如果我们通过对每一点的当前值、前导点和后续点计算“移动平均数”来修匀数据，常常能够使我们在参差不齐的序列中看到潜在的模式。比如，“跨距 3 的移动平均数”(moving average of span 3)就是指 y_{t-1}、y_t 和 y_{t+1} 的平均数。我们可以用 Stata 的明确下标来建立(**generate**)这样的变量：

```
. generate water3 = (water[_n-1] + water[_n] + water[_n+1])/3
```

或者，我们也能应用 **egen** 命令中的 **ma** 函数(即 moving average)来做：

```
. egen water3 = ma(water), nomiss t(3)
```

选项 **nomiss** 要求在序列两端计算较短跨距和不对中的移动平均数，否则新变量 *water3* 的第一个和最后一个值将为缺失。选项 **t(3)** 要求按跨距 3 来计算移动平均数，跨距可以定为任何大于等于 3 的奇数。

对于时间序列(**tsset**)数据，**tssmooth** 命令提供了多种强大的修匀工具。除 **tssmooth nl** 以外，所有修匀都能处理缺失值。

tssmooth ma	移动平均过滤器，可加权或不加权
tssmooth exponential	单指数过滤器
tssmooth dexponential	双指数过滤器
tssmooth hwinters	非季节性的 Holt-Winters 修匀
tssmooth shwinters	季节性的 Holt-Winters 修匀
tssmooth nl	非线性过滤器

键入 **help tssmooth_exponential**、**help tssmooth_hwinters** 等以咨询每种命令的语法。

图 13.2 画出了米尔福镇用水的简单 5 天移动平均数(*water5*)，同时展示了原始数据(*water*)。这一 **graph twoway** 命令将 *water5* 修匀值的绘线与 *water* 原始数据值的绘线(细线)重叠在一起了。横轴标签标志起始月份值是通过“手工”选择的(8401，8432 等)，以使图形更好理解。将标签格式化为**% dmd**(即“ 月 日”日期格式)。请比较图 13.2 的标签与其默认设置的图 13.1。

```
. tssmooth ma water5 = water, window(2 1 2)
The smoother applied was
     (1/5)*[x(t-2) + x(t-1) + 1*x(t) + x(t+1) + x(t+2)]; x(t)= water

. graph twoway line water5 date, clwidth(thick)
      ||  line water date, clwidth(thin) clpattern(solid)
      ||  , ylabel(300(100)900)
      xlabel(8401 8432 8460 8491 8521 8552 8582 8613,
         grid format(%dmd))
      xtitle("") ytitle(Water use in 1000 gallons)
      legend(order(2 1) position(4) ring(0) rows(2)
         label(1 "5-day average") label(2 "daily water use"))
```

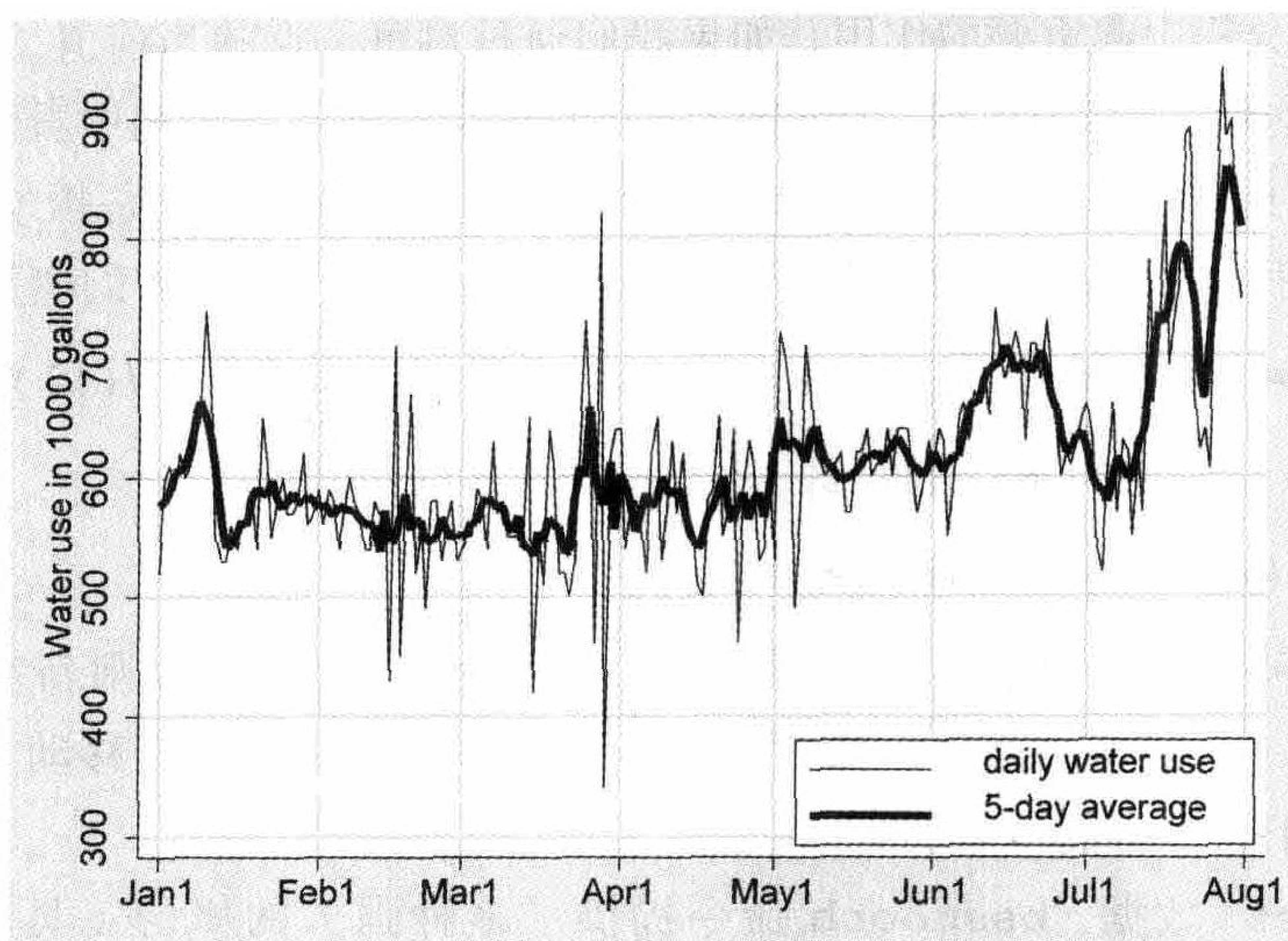


图 13.2

移动平均数也有其他基于平均数的统计量的共同缺点:它们对特异值没有抵抗力。由于特异值在图 13.1 中形成了许多突出的针尖,我们也可以尝试采用不同的修匀方法。命令 **tssmooth nl** 执行对特异值有抵抗力的非线性修匀,它所应用的方法以及有关术语可参见 Velleman 和 Hoaglin(1981)以及 Velleman(1982)的文献。比如:

```
. tssmooth nl water5r = water, smoother(5)
```

这个命令创建了一个名为 *water5r* 的新变量,保存按跨距 5 的移动中位数对 *water* 修匀后的值。可以按 Velleman 原始标注那样定义出复合校平器(compound smoother),采用不同跨距的移动中位数,再伴以"Hanning 加权函数"(即按跨距 3 进行 1/4、1/2、1/4 加权的移动平均数)和其他技术。有一种称为"4253h,twice"的复合校平器显得特别有用。将其用于 *water*,我们就计算出修匀变量 *water4r*:

```
. tssmooth nl water4r = water, smoother(4253h,twice)
```

图 13.3 画出了新的修匀值 *water4r*。比较图 13.3 与图 13.2 就看出这个"4253h,twice"修匀相对于一个移动平均修匀的功夫了。尽管两个校平器有着同样的跨距,但是"4253h,twice"修匀在减少参差不齐的变异上做得更多。

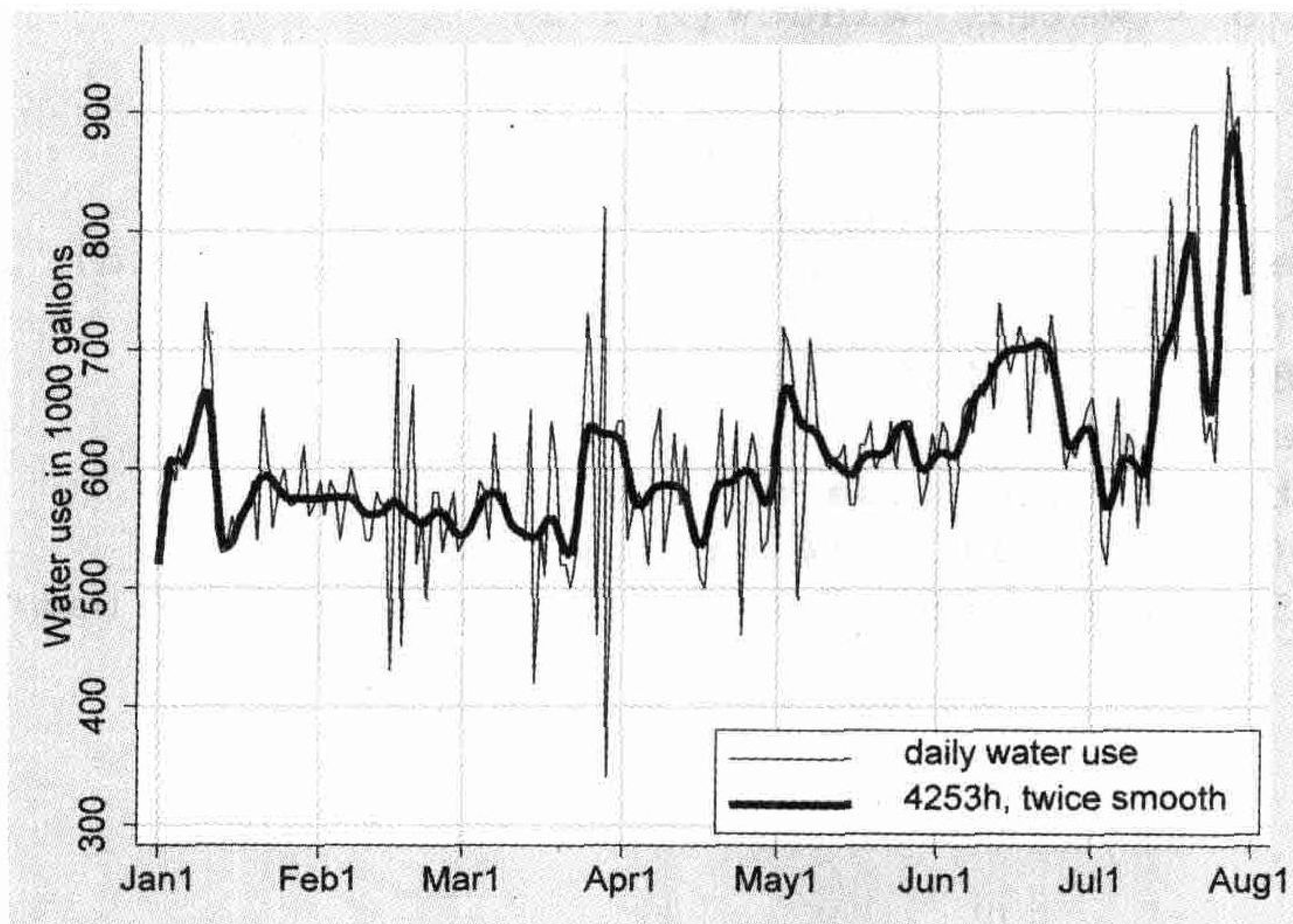


图 13.3

有时,我们修匀的目标是在修匀标绘图中寻找模式。然而,就这一特定数据而言,修匀以后的"粗糙"即残差实际上更有意思。我们可以计算出原始数据与修匀数据之间的差来作为其粗糙程度,然后将这些结果画成另一幅时间标绘图,图 13.4。

```
. generate rough = water - water4r
. label variable rough "Residuals from 4253h, twice"
. graph twoway line rough date,
      xlabel(8401 8432 8460 8491 8521 8552 8582 8613,
        grid format(%dmd)) xtitle("")
```

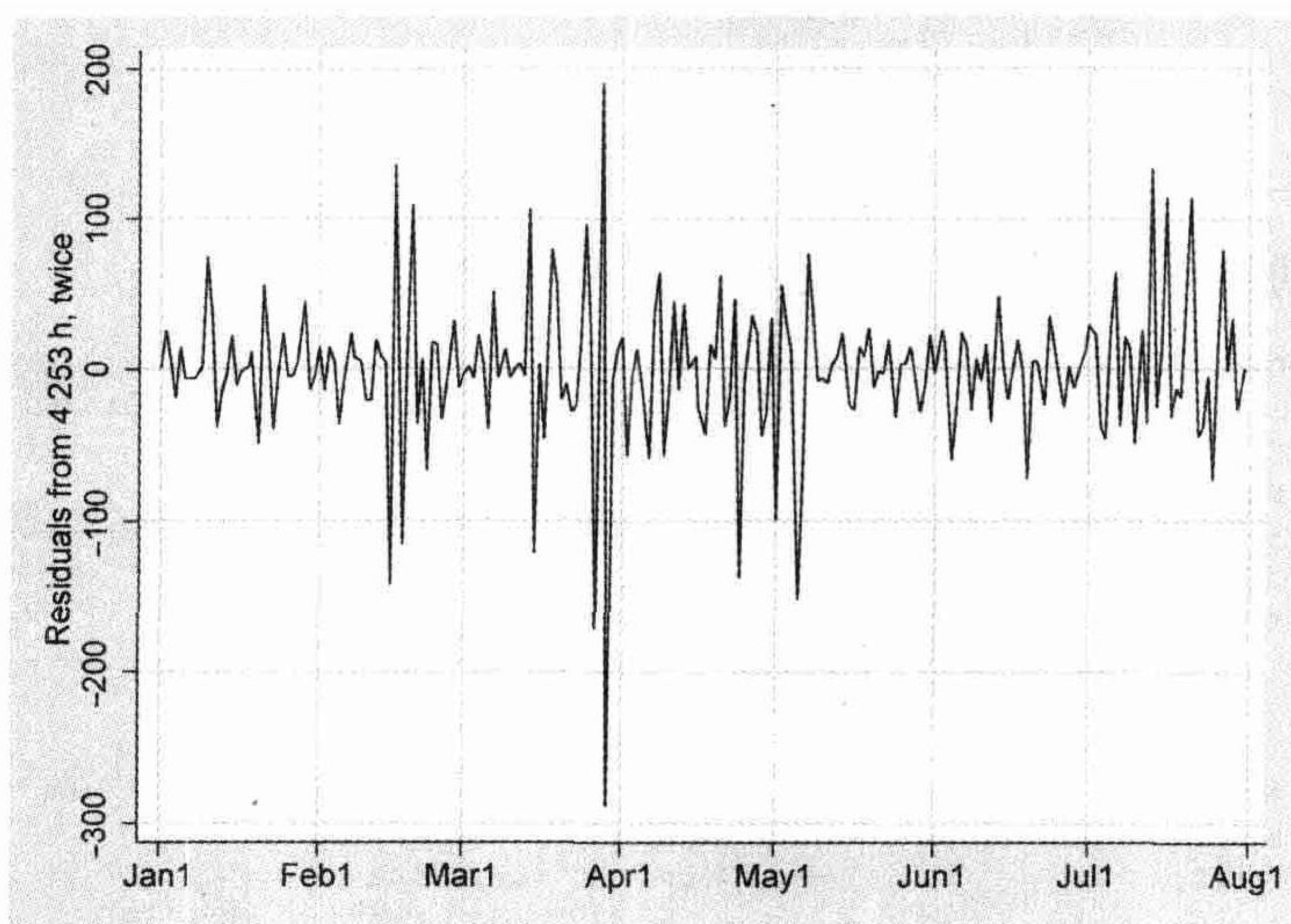


图 13.4

图 13.4 中最严重的波动发生在 3 月 27 至 29 日。用水量突然急剧下降,又重新升高,然后在恢复到通常水平之前下降到更低。就在这些日子里,本地报纸说,供应本地用水的一口井中发现了有害的化学废料。最初的报导警告了居民,并且后来在有问题的水井被查封以后又再次得到证实。

本节描述的修匀技术只有当观测在时间上具有相同间隔时才更有意义。如果时间序列有不一样的间隔,lowess 回归提供了一种实际的替代方法(参见第 8 章)。

更多时间标绘图例子

数据集 *atlantic.dta* 包括了北大西洋 1950—2000 年时间序列的气候、海洋和渔业变量(原始数据来源包括 Buch(2000),其他引用见 Hamilton、Brown 和 Rasmussen(2003))。变量中还有西格陵兰外菲拉堤岸(Fylla Bank)的海洋温度,格陵兰首府努克市的气温,两个气候指标称为北大西洋振荡(NAO)和北极振荡(AO),以及西格陵兰水域的鱼虾捕获量。

在分析这些时间序列数据以前,我们用 **tsset** 命令设置这一数据,告诉 Stata 变量 *year* 包含时间顺序的信息。

```
. tsset year, yearly

        time variable:  year, 1950 to 2000
```

在一个 **tsset** 数据中,有两种新的选择条件可用: **tin**(即在 [t1,t2]之间的时

间,times in)和 **twithin**(即在(t1,t2)之内的时间,times within)。要列出 1950—1955 年的菲拉温度和 NAO 值,键入:

```
. list year fylltemp wNAO if tin(1950,1955)

     +-------------------------+
     | year   fylltemp    wNAO |
     |-------------------------|
  1. | 1950        2.1     1.4 |
  2. | 1951        1.9   -1.26 |
  3. | 1952        1.6     .83 |
  4. | 1953        2.1     .18 |
  5. | 1954        2.3     .13 |
     |-------------------------|
  6. | 1955        1.2   -2.52 |
     +-------------------------+
```

twithin 选择条件的使用十分类似,但是选择不包括两个端点:

```
Contains data from atlantic.dta
  obs:            51                          Greenland climate & fisheries
 vars:             8                          27 Jul 2005 12:41
 size:          1734 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %ty                    Year
fylltemp        float  %9.0g                  Fylla Bank temp. at 0-40m
fyllsal         float  %9.0g                  Fylla Bank salinity at 0-40m
nuuktemp        float  %9.0g                  Nuuk air temperature
wNAO            float  %9.0g                  Winter (Dec-Mar)
                                                Lisbon-Stykkisholmur NAO
wAO             float  %9.0g                  Winter (Dec-Mar) AO index
tcod1           float  %9.0g                  Division 1 cod catch, 1000t
tshrimp1        float  %9.0g                  Division 1 shrimp catch, 1000t
-------------------------------------------------------------------------------
Sorted by:  year
```

```
. list year fylltemp wNAO if twithin(1950,1955)

     +-------------------------+
     | year   fylltemp    wNAO |
     |-------------------------|
  2. | 1951        1.9   -1.26 |
  3. | 1952        1.6     .83 |
  4. | 1953        2.1     .18 |
  5. | 1954        2.3     .13 |
     +-------------------------+
```

我们用 **tssmooth nl** 来定义一个新变量 *fy114*,存放“4253h, twice”对 *fylltemp*(数据来自 Buch,2000)的修匀值。

```
. tssmooth nl fy114 = fylltemp, smoother(4253h, twice)
```

图 13.5 绘出了菲兰堤岸水温的原始值(*fylltemp*)和修匀值(*fy114*)。原始水温显示为平均数(1.67 ℃)离差的芒线图,所以这个图强调了 10 年周期和每年的变化。

```
. graph twoway spike fylltemp year, base(1.67) yline(1.67)
     ||  line fy114 year, clpattern(solid)
     ||  , ytitle("Fylla Bank temperature, degrees C") ylabel(0(1)3)
      xtitle("") xtick(1955(10)1995) legend(off)
```

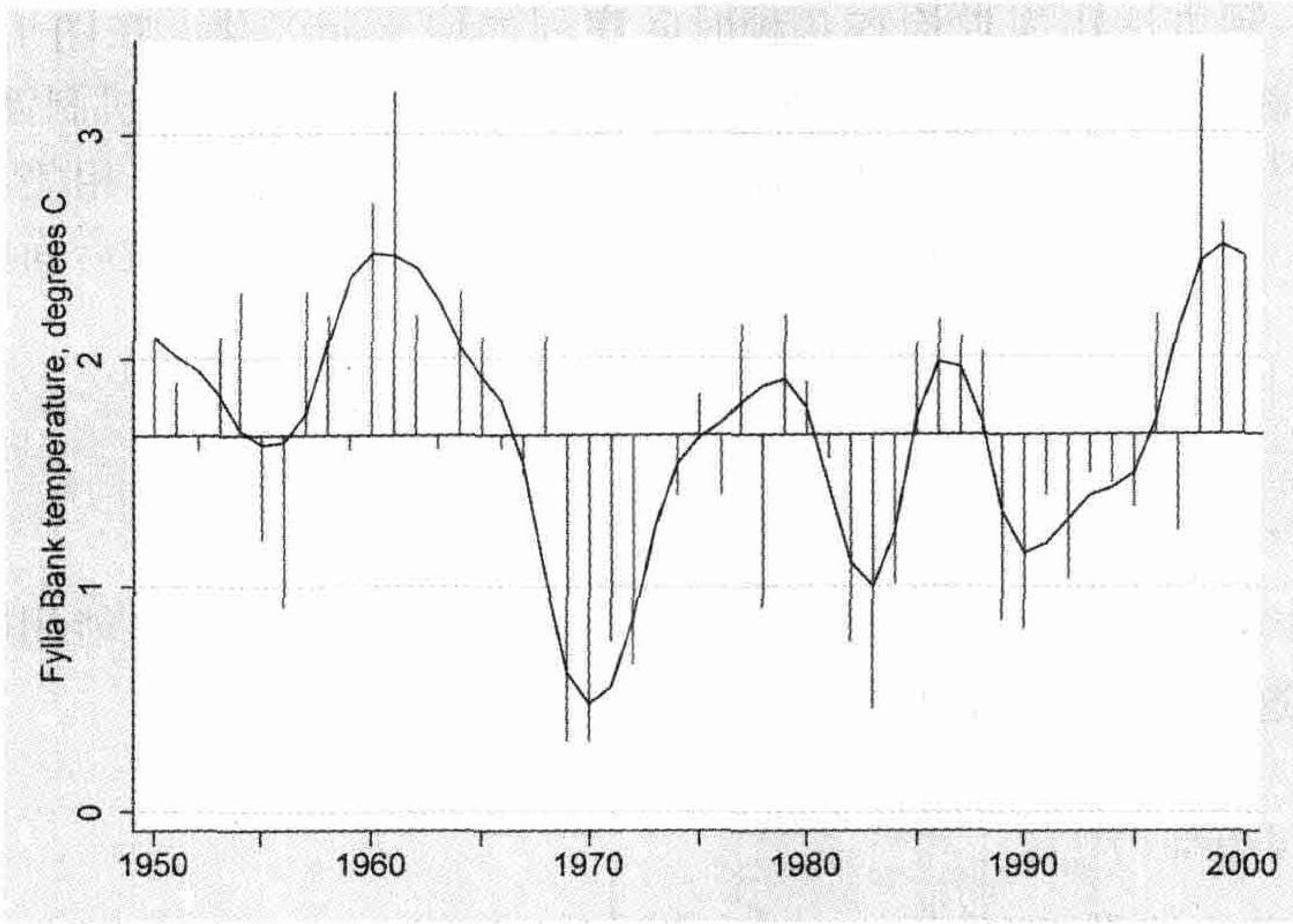


图 13.5

图 13.5 中的修匀值展示了水温一般从较暖到较冷的时期并不规则。当然,"较暖"是相对于格陵兰而言的,其夏天的水温也从未高过 3.34 ℃(37 ℉)。

菲兰堤岸的温度受大规模大气模式的影响,这种大气模式称为北大西洋振荡或 NAO。图 13.6 画出了修匀温度,同时画出了 NAO 的修匀值(新变量 *wNAO4*)。在这一重叠图中,左轴即 **yaxis(1)** 为温度,右轴即 **yaxis(2)** 为 NAO。其他 y 轴选项进一步指定是参照轴 1 还是参照轴 2。比如,**yline(0, axis(2))** 画出的水平线标志了 NAO 指标的 0 点。在两边的坐标上,数值标签都是水平排印的。图例按 5 点钟位置显示于图中空间里,选项为 **position(5) ring(0)**。

```
. graph twoway line fyll4 year, yaxis(1)
     ylabel(0(1)3, angle(horizontal) nogrid axis(1))
     ytitle("Fylla Bank temperature, degrees C", axis(1))
     || line wNAO4 year, yaxis(2) ytitle("Winter NAO index", axis(2))
     ylabel(-3(1)3, angle(horizontal) axis(2)) yline(0, axis(2))
     ||  , xtitle("") xlabel(1950(10)2000, grid) xtick(1955(5)1995)
     legend(label(1 "Fylla temperature") label(2 "NAO index") cols(1)
        position(5) ring(0))
```

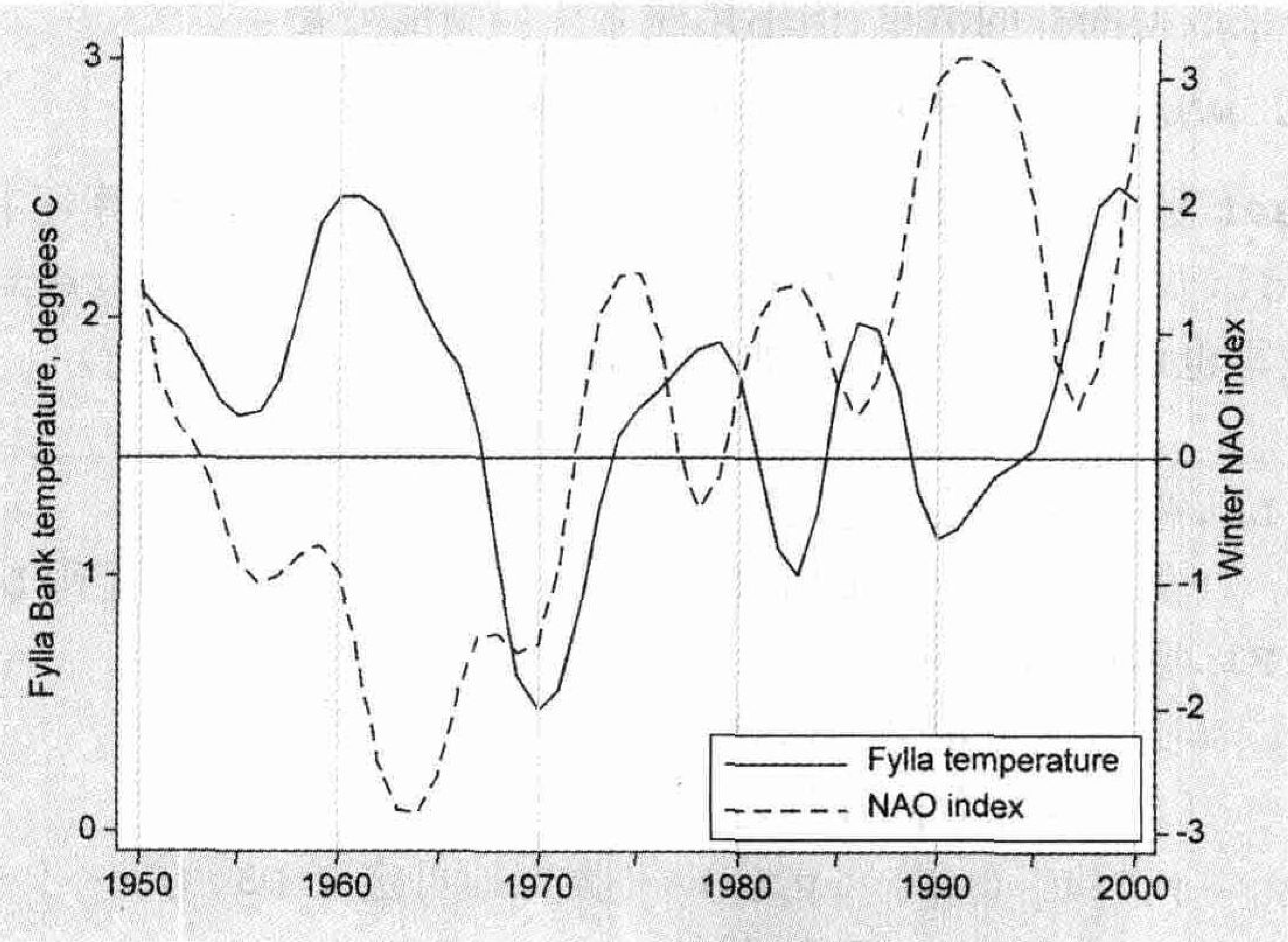


图 13.6

重迭标绘图提供了一种方法用视觉检查几种时间序列如何一起变化。在图 13.6 中,我们看到了负相关的迹象:NAO 高时对应着低温度。这种相关背后的物理机制涉及,在高 NAO 阶段,北风将北极空气和海水带到西格陵兰。温度与 NAO 的这种负相关在这一时间序列数据的后面一段变得更强,大概在 1973—1997 年期间。我们将在后面再来讨论这种关系。

时滞、前导和差分

时间序列分析经常涉及时滞变量,或者说就是前次观测的值。时滞(lag)能用明确下标来定义。比如,下面这个命令创建的变量 *wNAO_1* 等于前一年的 NAO 值:

```
. generate wNAO_1 = wNAO[_n-1]
(1 missing value generated)
```

另一种方式也能取得同样的结果,即采用 **tsset** 数据时,再加上 Stata 的 **L.**(表示 lag)运算符:

```
. generate wNAO_1 = L.wNAO
(1 missing values generated)
```

时滞运算经常要比明确下标的方法更为简单。更重要的是,时滞运算也能用于面板数据。要建立时滞 2 的值,可用以下命令:

```
. generate wNAO_2 = L2.wNAO
(2 missing values generated)

. list year wNAO wNAO_1 wNAO_2 if tin(1950,1954)

     +-----------------------------------+
     | year    wNAO    wNAO_1    wNAO_2  |
     |-----------------------------------|
  1. | 1950     1.4         .         .  |
  2. | 1951   -1.26       1.4         .  |
  3. | 1952     .83     -1.26       1.4  |
  4. | 1953     .18       .83     -1.26  |
  5. | 1954     .13       .18       .83  |
     +-----------------------------------+
```

我们还可以用另一种命令来取得同样的清单、但不用建立任何新的变量:

```
. list year wNAO L.wNAO L2.wNAO if tin(1950,1954)
```

L.运算只是简化 **tsset** 数据运算的几种方法之一。其他的时间序列运算还有 **F.**(前导,lead)、**D.**(差分,difference)以及 **S.**(季节差分,seasonal difference)。这些运算符既可以采用大写也可以采用小写,比如,**F2.*wNAO*** 或 **f2.*wNAO*** 都行。

时间序列运算

L.	时滞 y_{t-1} (**L1.**意味着同样的定义)
L2.	2 期时滞 y_{t-2} (类似地可定义 **L3.**等。**L(1/4).**则定义从 **L1.**直到 **L4.**)
F.	前导 y_{t+1} (**F1.**意味着同样的定义)
F2.	2 期前导 y_{t+2} (类似地可定义 **F3.**等)
D.	差分 $y_t - y_{t-1}$ (**D1.**意味着同样的定义)
D2.	2 阶差分 $(y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$ (类似地可定义 **D3.**等)
S.	季节差分 $y_t - y_{t-1}$ (它与 **D.**定义相同)
S2.	2 期季节差分 $y_t - y_{t-2}$ (类似地可定义 **S3.**等)

在季节差分的情况下,**S12.**并不意味着"12 阶差分",而是指时滞为 12 期的一阶差分。比如,如果我们有按月的温度而不是按年的温度,我们可能想计算 **s12.*temp***,那么它将是2000 年12 月温度与1999 年12 月温度之差,如此等等。

时滞运算可以直接置入大多数分析命令中。我们可以将 1973—1997 年的 *fylltemp* 对修匀变量 *wNAO* 做回归,再加上时滞分别为 1、2、3 年的自变量 *wNAO*,这并不需要事先创建任何时滞变量。

```
. regress fylltemp wNAO L1.wNAO L2.wNAO L3.wNAO if tin(1973,1997)

      Source |       SS       df       MS              Number of obs =      25
-------------+------------------------------           F(  4,    20) =    4.57
       Model |  3.1884913     4  .797122826            Prob > F      =  0.0088
    Residual | 3.48929123    20  .174464562            R-squared     =  0.4775
-------------+------------------------------           Adj R-squared =  0.3730
       Total | 6.67778254    24  .278240939            Root MSE      =  .41769

------------------------------------------------------------------------------
fylltemp     |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
wNAO         |
          -- |  -.1688424   .0412995    -4.09   0.001    -.2549917   -.0826931
          L1 |   .0043805   .0421436     0.10   0.918    -.0835294    .0922905
          L2 |  -.0472993    .050851    -0.93   0.363    -.1533725     .058774
          L3 |   .0264682   .0495416     0.53   0.599    -.0768738    .1298102
_cons        |   1.727913   .1213588    14.24   0.000     1.474763    1.981063
-------------+----------------------------------------------------------------
```

与此等价,我们也可以键入以下命令:

```
. regress fylltemp L(0/3).wNAO if tin(1973,1997)
```

所估计的模型为

预测值:$fylltemp_t = 1.728 - 0.169\,wNAO_t + 0.004\,wNAO_{t-1} - 0.047\,wNAO_{t-2} + 0.026\,wNAO_{t-3}$

所有时滞项的系数统计性都不显著,看起来 *wNAO* 的当前值(非时滞)提供了最简约的预测。的确,如果我们再重新估计这个模型并取消那些时滞变量,那么调整的(adjusted)R^2就从原来的 0.37 提高到 0.43。然而,这两个模型都是很粗糙的。对自相关误差的 Durban-Watson 检验虽然也是不确定的,但是在这样的小样本上它并不太可靠。

```
. dwstat

Durbin-Watson d-statistic(  5,    25) =  1.423806
```

在时间序列分析中常常遇到自相关误差,它的存在通常使 OLS 估计的置信区间和统计检验无效。对时间序列更恰当的回归方法将在本章后面加以讨论。

相关图

自相关系数用于估计一个变量与其自身某一时滞之间的相关。比如,一阶自相关(first-order autocorrelation)是 y_t 和 y_{t-1} 之间的相关。二阶自相关则是指 $Cor[y_t, y_{t-2}]$,以此类推。相关图(correlogram)可以画出相关与时滞之间的关系。

Stata 的 **corrgram** 命令提供了简单的相关图和有关的信息。它所显示的最大时滞是由数据所制约的,可以用 **matsize** 调用最大时滞,或者用 **lags( )**选项来指定任意的较小值:

```
. corrgram fylltemp, lags(9)

                                          -1       0       1 -1       0       1
 LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1       0.4038   0.4141   8.8151  0.0030          |---               |---
2       0.1996   0.0565   11.012  0.0041          |-                 |
3       0.0788   0.0045   11.361  0.0099          |                  |
4       0.0071  -0.0556   11.364  0.0228          |                  |
5      -0.1623  -0.2232   12.912  0.0242         -|                 -|
6      -0.0733   0.0880   13.234  0.0395          |                  |
7       0.0490   0.1367   13.382  0.0633          |                  |-
8      -0.1029  -0.2510   14.047  0.0805          |                --|
9      -0.2228  -0.2779   17.243  0.0450         -|                --|
```

时滞(LAG)列在表的左边,接着列有自相关(AC)和偏自相关(PAC)。比如,在 $fylltemp_t$和 $fylltemp_{t-2}$之间的相关为0.199 6,相应的偏自相关(已经调整了时滞1)为0.056 5。Q统计量(即Box-Pierce混合法)检验的是一系列虚无假设,即所有各种时滞的自相关都为0。因为这里看到的绝大多数 P 值都低于0.05,我们能够拒绝这些虚无假设,认为 *fylltemp* 存在显著的自相关。如果 Q 统计量的 P 值不低于0.05,我们就能够肯定这个序列是不含显著自相关的"白噪声"(white noise)。

在这一输出的右侧是用字符构成的自相关与偏自相关的标绘图。审查这种图对于设置时间序列模型很重要。更精细的自相关标绘图可以通过 **AC** 命令来取得:

```
. ac fylltemp, lags(9)
```

得到的相关图,图13.7,包括了95%置信区间的阴影区域标注。在这一区间之外的那些相关都是个体显著的。

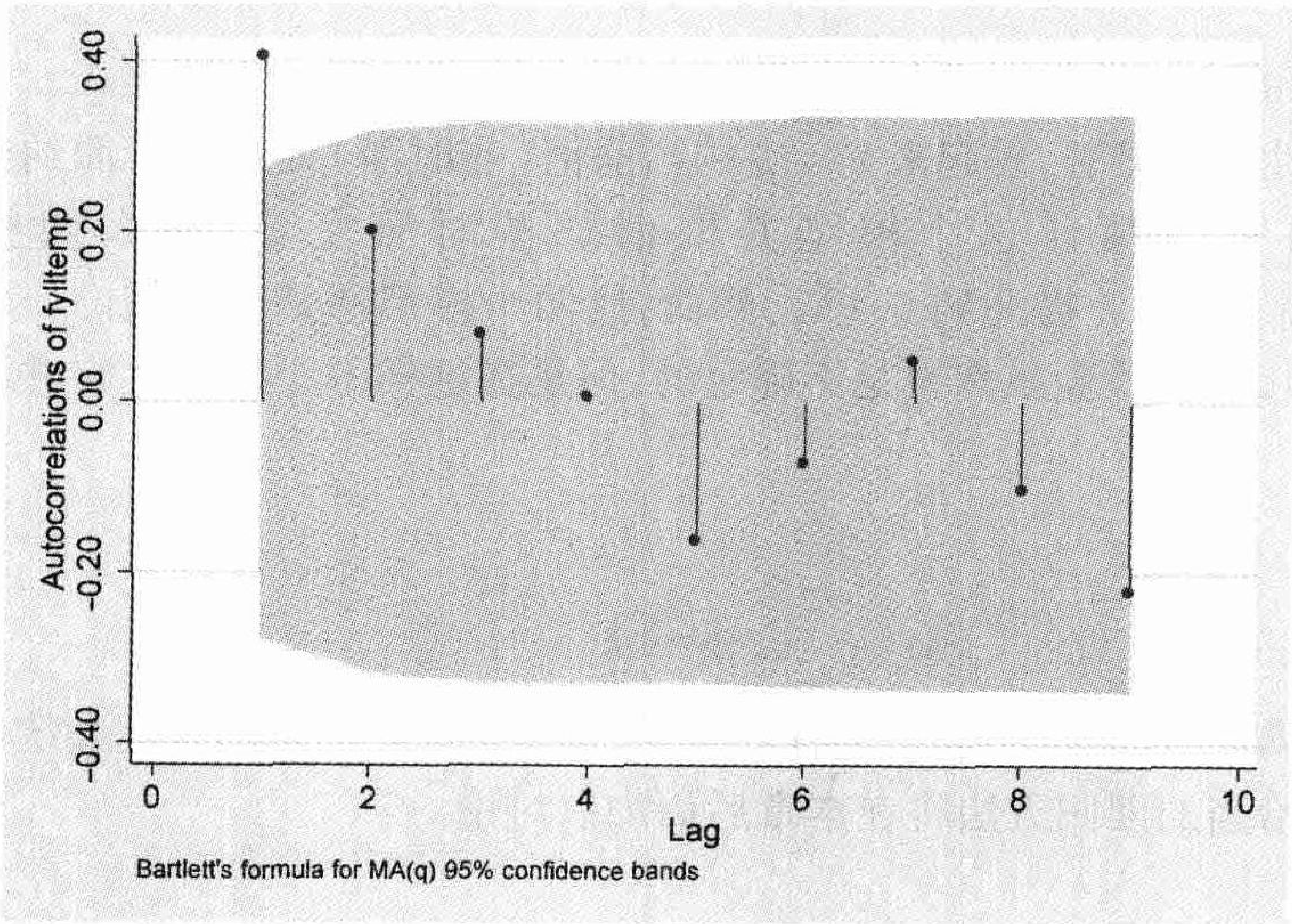


图13.7

一个类似的命令 **PAC** 可以提供图13.8那样的偏自相关图。近似置信区间(将标准误按 $1/\sqrt{n}$来估计)也显示在图13.8中。命令 **AC** 和 **PAC** 的默认图形就像图13.7那样。而对于图13.8,我们选择了不同选项,绘出了0相关的基准线,还将置信区间表示成轮廓、而不是阴影区域。

```
. pac fylltemp, yline(0) lags(9) ciopts(bstyle(outline))
```

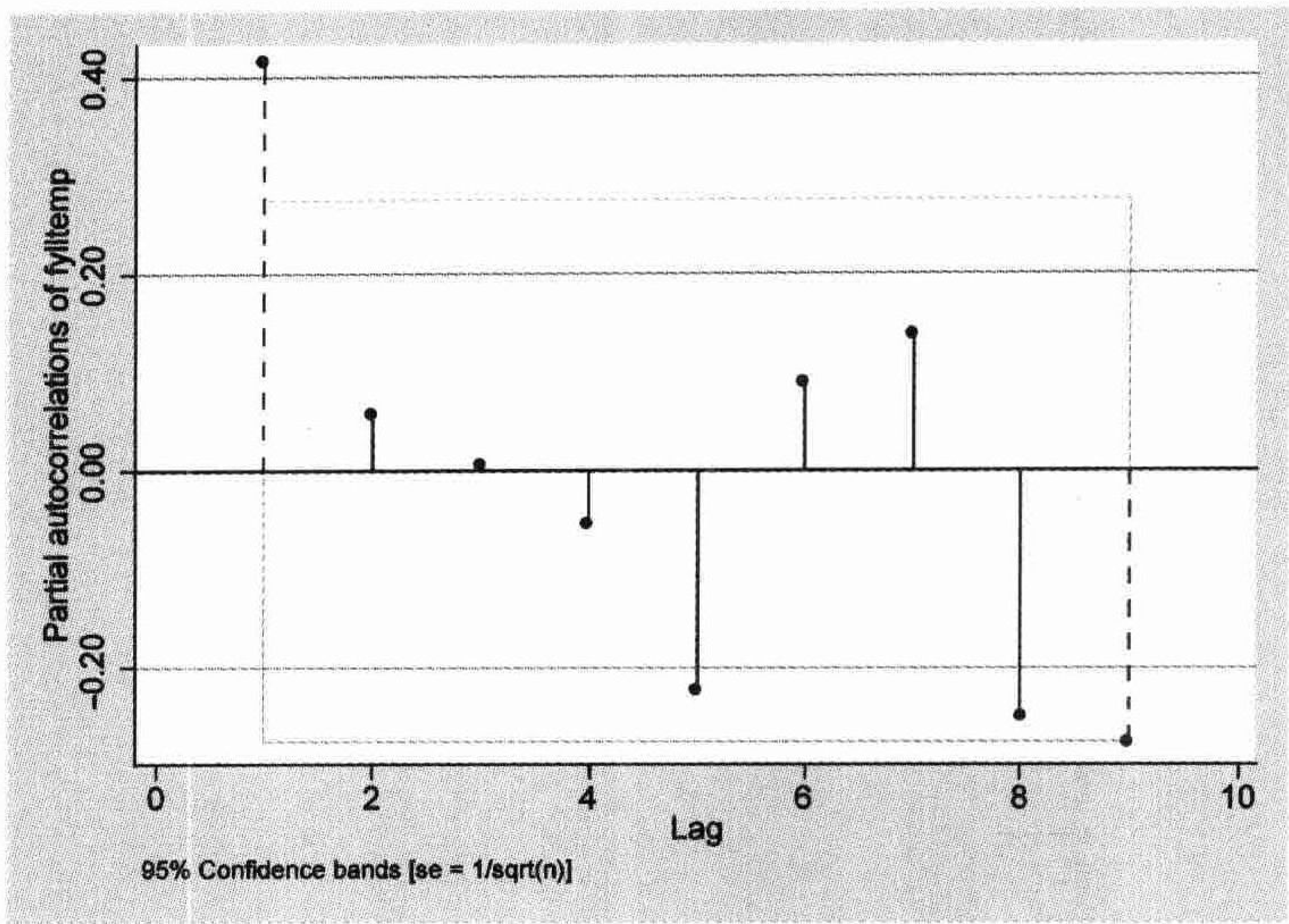


图 13.8

交叉相关图有助于探测两个时间序列之间的关系。图 13.9 显示了 *wNAO* 和 *fylltemp* 在 1973—1997 年之间的交叉相关图。在时滞为 0 时的交叉相关很强，而且是负相关，但是在其他正的或负的时滞时，交叉相关接近于 0。这表明，这两个序列之间的关系是"即时的(instantanious)"(在每年数据中)，而不是滞后的或是分布于好几年中的。请回忆，我们前面所做的 OLS 回归中，时滞自变量就都不显著。

```
. xcorr wNAO fylltemp if tin(1973,1997), lags(9) xlabel(-9(1)9, grid)
```

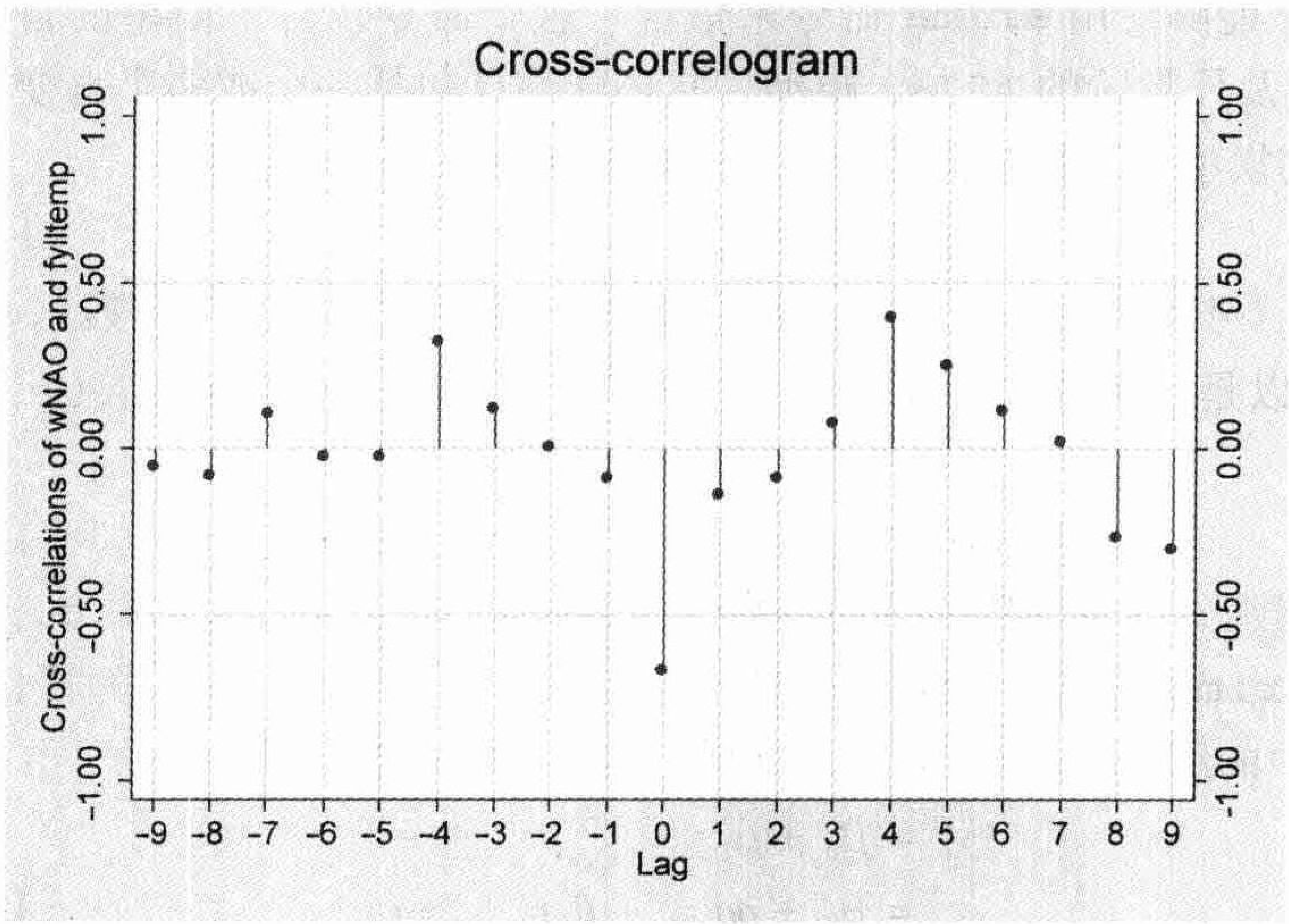


图 13.9

如果我们在 **xcorr** 命令中先列出自变量、后列出因变量，就像在图 13.9 中那样，那么正的时滞表明 t 时的自变量与 $t+1$、$t+2$……时的因变量之间相关。于是，我们看到在冬季的 NAO 指标与 4 年以后的菲拉温度之间的相关为 0.394。

实际交叉相关系数以及文本版的交叉相关图能够通过加上 **table** 选项来取得：

```
. xcorr wNAO fylltemp if tin(1973,1997), lags(9) table
```

```
                        -1          0          1
 LAG        CORR       [Cross-correlation]
-----------------------------------------------
-9         -0.0541                  |
-8         -0.0786                  |
-7          0.1040                  |
-6         -0.0261                  |
-5         -0.0230                  |
-4          0.3185                  |--
-3          0.1212                  |
-2          0.0053                  |
-1         -0.0909                  |
0          -0.6740             -----|
1          -0.1386                 -|
2          -0.0865                  |

3           0.0757                  |
4           0.3940                  |---
5           0.2464                  |-
6           0.1100                  |
7           0.0183                  |
8          -0.2699                --|
9          -0.3042                --|
```

ARIMA 模型

时间序列数据的自回归集成移动平均法(ARIMA,autoregressive integrated moving average)模型能够使用 **arima** 命令来估计。这个命令包含了简单的自回归(AR)、移动平均(MA)以及任意阶的 ARIMA 模型。它还能估计包括一个或多个自变量以及 AR 或 MA 误差的结构模型。这种结构模型的通用形式用矩阵来表示就是:

$$y_t = x_t\beta + \mu_t \quad [13.1]$$

其中 y_t 为因变量向量在 t 时的取值,x_t 是自变量值的矩阵(通常还包括一个常数),μ_t 是扰动向量。这些扰动可以是任意阶上的自回归或移动平均的扰动。比如,ARMA(1,1)[17] 的扰动为:

$$\mu_t = \rho\mu_{t-1} + \theta \in_{t-1} + \in_t \quad [13.2]$$

其中 ρ 是一阶自相关参数,θ 是一阶移动平均参数,$\in$ 是白噪声(*normal i.i.d.*,即正态独立同分布)扰动。**arima** 可以拟合简单模型,作为公式[13.1]和[13.2]的特例,用一个常数(β_0)代替其中的结构项 $X_t\beta$。所以,简单的 ARMA(1,1)模型成为:

$$\begin{aligned} y_t &= \beta_0 + \mu_t \\ &= \beta_0 + \rho\mu_{t-1} + \theta \in_{t-1} + \in_t \end{aligned} \quad [13.3]$$

某些来源提供了一种不同的版本。在 ARMA(1,1)的情况下,他们将 y_t 作为以前 y 值(即 y_{t-1})和当前扰动($\in_t$)以及时滞扰动($\in_{t-1}$)的函数:

$$y_t = \alpha + \rho y_{t-1} + \theta \in_{t-1} + \in_t \quad [13.4]$$

因为在这种简单结构模型 $y_t = \beta_0 + \mu_t$ 中,公式[13.3](Stata 的版本)与公式[13.4]等价,只是改变了常数尺度,即 $\alpha = (1-\rho)\beta_0$。

使用 **arima** 时,一个 ARMA(1,1)模型(即公式[13.3])可以任用以下两种方式之一:

---

[17]【译注:ARMA 是自回归移动平均法的缩写,即 autoregressive moving average。】

```
. arima y, ar(1) ma(1)
```

或

```
. arima y, arima(1,0,1)
```

arima 中的 **i** 代表“集成”(integrated),指那些还涉及差分的模型。要拟合一个 ARIMA(2,1,1)模型,使用以下命令:

```
. arima y, arima(2,1,1)
```

或者用等价的命令:

```
. arima D.y, ar(1 2) ma(1)
```

这两个命令定义了一样的模型,其中因变量的一阶差分($y_t - y_{t-1}$)是前两个时滞的一阶差分($y_{t-1} - y_{t-2}$和 $y_{t-2} - y_{t-3}$)和当前与以前扰动(ϵ_t 和 ϵ_{t-1})的函数。

要估计一个其中 y_t 依赖于两个自变量 x(当前值和时滞值,即 x_t 和 x_{t-1})和 w(只有当前值 w_t)、并包含 ARIMA(1,0,1)误差的结构模型,恰当的命令为:

```
. arima y x L.x w, arima(1,0,1)
```

尽管季节差分(比如,**S12.*y***)和(或)季节时滞(比如,**L12.*x***)也可以包括在内,但是按以上所写的命令,**arima** 并不估计乘积型 ARIMA$(p,d,q)(P,D,Q)_s$的季节模型。

当时间序列 y 的平均数和方差不随时间变化,并且当 y_t 和 y_{t+u}之间的协方差只依赖于时滞 u 而并不依赖于某一特定 t 值,那么这个时间序列 y 就被认为是“稳态”(stationarity)。ARIMA 建模时假定,我们的序列是稳态的,或者可以通过适当的差分或转换形成稳态。我们可以通过审视时间标绘图中水平或方差的趋势来非正规地检查这一假定。对“单位根”(unit root)(一个非稳态的 AR(1)过程的$\rho_1 = 1$,也被称为“随机散步”)的正规统计检验也有所帮助。Stata 提供三种单位根检验:**pperron**(即 Phillips-Perron 检验),**dfuller**(即扩展的 Dickey-Fuller 检验),以及 **dfgls**(应用 GLS 的扩展的 Dickey-Fuller 检验,通常是比 **dfuller** 更为强大的检验)。

应用到非拉堤岸温度数据,**pperron** 检验拒绝了单位根的虚无假设($P<0.01$)。

```
. pperron fylltemp, lag(3)

Phillips-Perron test for unit root                 Number of obs   =        50
                                                   Newey-West lags =         3

                           ---------- Interpolated Dickey-Fuller ---------
                  Test         1% Critical       5% Critical      10% Critical
               Statistic           Value             Value             Value
------------------------------------------------------------------------------
 Z(rho)          -29.871           -18.900           -13.300           -10.700
 Z(t)             -4.440            -3.580            -2.930            -2.600
------------------------------------------------------------------------------
* MacKinnon approximate p-value for Z(t) = 0.0003
```

与此类似,Dickey-Fuller 的 GLS 检验评价的虚无假设为 *fylltemp* 有一个单位根(对应的替换假设为:它是稳态的,有一个很可能为非 0 的平均数,但是没有线性的时间趋势),检验拒绝了虚无假设($P<0.05$)。于是,两个检验都肯定了从图 13.5 获得的稳态视觉印象。

```
. dfgls fylltemp, notrend maxlag(3)
```

```
DF-GLS for fylltemp           Number of obs =     47

               DF-GLS mu      1% Critical     5% Critical     10% Critical
  [lags]    Test Statistic       Value           Value            Value
------------------------------------------------------------------------------
    3           -2.304          -2.620          -2.211           -1.913
    2           -2.479          -2.620          -2.238           -1.938
    1           -3.008          -2.620          -2.261           -1.959

Opt Lag (Ng-Perron seq t) = 0 [use maxlag(0)]
Min SC   = -.6735952 at lag  1 with RMSE  .6578912
Min MAIC = -.2683716 at lag  2 with RMSE  .6569351
```

对于一个稳态序列，相关图提供了关于选择一个初步的 ARIMA 模型的指导：

AR(p)　　一个 p 阶自回归过程存在自相关，它随着时滞的增加而逐渐衰减。在时滞 p 以后，偏自相关就断绝了。

MA(q)　　一个 q 阶移动平均过程存在自相关，在时滞 q 以后，自相关就断绝了。偏自相关随着时滞的增加而逐渐衰减。

ARMA(p,q)　　一个混合的自回归和移动平均过程存在自相关和偏自相关，随着时滞的增加而逐渐衰减。

相关图中在季节时滞上的尖芒（比如，在按月的数据中的12、24、36号）表明了季节性模式。辨认季节模型服从于类似的指导方针，但是要在自相关和偏自相关分析时按照季节时滞。

图13.7 和图13.8 微弱地表现为是个AR(1)过程，所以我们将其作为 *fylltemp* 的简单模型来试一试。

```
. arima fylltemp, arima(1,0,0) nolog

ARIMA regression

Sample:  1950 to 2000                           Number of obs      =        51
                                                Wald chi2(1)       =      7.53
Log likelihood = -48.66274                      Prob > chi2        =    0.0061
------------------------------------------------------------------------------
             |                 OPG
fylltemp     |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
fylltemp     |
_cons        |    1.68923   .1513096    11.16   0.000     1.392669    1.985792
-------------+----------------------------------------------------------------
ARMA         |
ar           |
          L1 |   .4095759   .1492491     2.74   0.006     .1170531    .7020987
-------------+----------------------------------------------------------------
      /sigma |    .627151   .0601859    10.42   0.000     .5091889    .7451131
------------------------------------------------------------------------------
```

在我们拟合了一个 **arima** 模型以后，它的系数和其他结果以 Stata 通常的方式被暂时保存起来。比如，要想看看最近的 AR(1)模型的系数和标准误，可以键入：

```
. display [ARMA]_b[L1.ar]
.4095759

. display [ARMA]_se[L1.ar]
.14924909
```

这一例子的 AR(1)系数统计性显著地区别于0（$z = 2.74$，$p = 0.006$），提供了模型恰当的一个迹象。第二个检验是残差是否表现为不相关的“白噪声”。在执行了 **arima** 以后，我们可以通过 **predict** 来取得残差（也可取得预测值和其他案例统计量）：

```
. predict fyllres, resid
. corrgram fyllres, lags(15)

                                          -1       0       1 -1       0       1
 LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1        -0.0173  -0.0176   .0162  0.8987          |                 |
2         0.0467   0.0465  .13631  0.9341          |                 |
3         0.0386   0.0497  .22029  0.9742          |                 |
4         0.0413   0.0496  .31851  0.9886          |                 |
5        -0.1834  -0.2450  2.2955  0.8069         -|                -|
6        -0.0498  -0.0602  2.4442  0.8747          |                 |
7         0.1532   0.2156  3.8852  0.7929          |-                |-
8        -0.0567  -0.0726   4.087  0.8492          |                 |
9        -0.2055  -0.3232  6.8055  0.6574         -|               --|
10       -0.1156  -0.2418  7.6865  0.6594          |                -|
11        0.1397   0.2794  9.0051  0.6214          |-                |--
12       -0.0028   0.1606  9.0057  0.7024          |                 |-
13        0.1091   0.0647  9.8519  0.7060          |                 |
14        0.1014  -0.0547  10.603  0.7169          |                 |
15       -0.0673  -0.2837  10.943  0.7566          |               --|
```

corrgram 的 *Q* 检验发现,直到时滞 15,残差中并没有显著自相关。我们还能通过对时滞 15 做一次 **wntestq**(即用 *Q* 统计量检验白噪声,white noise test *Q* statistics)也能得到同样结果。

```
. wntestq fyllres, lags(15)

Portmanteau test for white noise
---------------------------------------
 Portmanteau (Q) statistic =    10.9435
 Prob > chi2(15)           =     0.7566
```

根据这些标准,我们的 AR(1)或 ARIMA(1,0,0)模型显得是恰当的。用 MA 或更高阶的 AR 项的更复杂的模型并不会对拟合有什么改进。

用一个类似的 AR(1)模型只拟合 1973—1997 年期间的 *fylltemp*。然而,在这一期间,冬季北大西洋振荡(*wNAO*)的信息显著地改进了预测。对于这个模型,我们将 *wNAO* 作为自变量,但是保留一个 AR(1)项来解释误差的自相关。

```
. arima fylltemp wNAO if tin(1973,1997), ar(1) nolog

ARIMA regression

Sample:  1973 to 1997                          Number of obs      =        25
                                               Wald chi2(2)       =     12.73
Log likelihood =  -10.3481                     Prob > chi2        =    0.0017

------------------------------------------------------------------------------
             |                 OPG
fylltemp     |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
fylltemp     |
wNAO         |  -.1736227   .0531688    -3.27   0.001    -.2778317   -.0694138
_cons        |   1.703462   .1348599    12.63   0.000     1.439141    1.967782
-------------+----------------------------------------------------------------
ARMA         |
ar           |
          L1 |   .2965222    .237438     1.25   0.212    -.1688478    .7618921
-------------+----------------------------------------------------------------
      /sigma |     .36536   .0654008     5.59   0.000     .2371767    .4935432
------------------------------------------------------------------------------
```

```
. predict fyllhat
(option xb assumed; predicted values)

. label variable fyllhat "predicted temperature"

. predict fyllres2, resid

. corrgram fyllres2, lags(9)

                                          -1       0       1 -1       0       1
LAG       AC       PAC       Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1      0.1485   0.1529   1.1929   0.2747          |-                  |-
2     -0.1028  -0.1320   1.7762   0.4114          |                  -|
3      0.0495   0.1182   1.9143   0.5904          |                   |
4      0.0887   0.0546   2.3672   0.6686          |                   |
5     -0.1690  -0.2334   4.0447   0.5430         -|                  -|
6     -0.0234   0.0722   4.0776   0.6662          |                   |
7      0.2658   0.3062   8.4168   0.2973          |--                 |--
8     -0.0726  -0.2236   8.7484   0.3640          |                  -|
9     -0.1623  -0.0999   10.444   0.3157         -|                   |
```

在这个模型中 *wNAO* 有显著的负系数。而 AR(1)系数的统计性却并不统计显著。然而，如果我们排除这个 AR 项，我们的残差便不能通过 **corrgram**(相关图)对白噪声的检验。图 13.10 同时画出了预测值 *fyllhat* 和观测温度序列 *fylltemp*。这个模型在拟合主要的变暖至变冷段落和几个较小变动方面做得相当不错。为了让图中的 *y* 轴标签显示为同样的小数位(即用 0.5、1.0、1.5 等代替 0.5、1、1.5 等)，我们指定它们的格式为**% 2.1f**。

```
. graph twoway line fylltemp year if tin(1973, 1997)
      || line fyllhat year if tin(1973, 1997)
      || , ylabel(.5(.5)2.5, angle(horizontal) format(%2.1f))
      ytitle("Degrees C") xlabel(1975(5)1995, grid) xtitle("")
      legend(label(1 "observed temperature")
         label(2 "model prediction") position(5) ring(0) col(1))
```

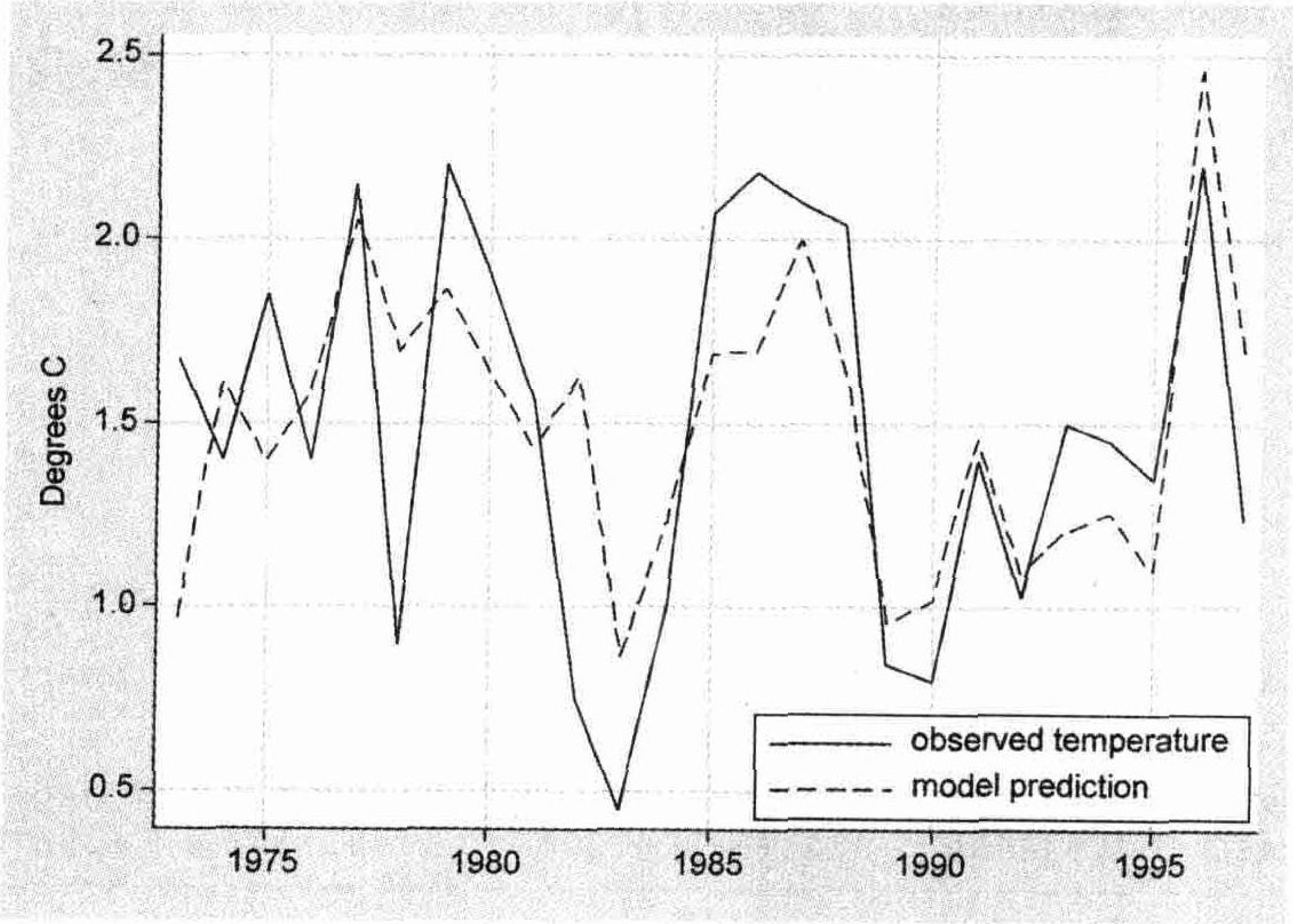


图 13.10

有一种称为 Prais-Winsten 回归(**Prais**)的技术，它用于修正一阶自回归误差，也可以用这个例子来加以示范。

```
. prais fylltemp wNAO if tin(1973,1997), nolog

Prais-Winsten AR(1) regression -- iterated estimates

      Source |       SS       df       MS              Number of obs =      25
-------------+------------------------------           F(  1,    23) =   23.14
       Model |  3.35819258     1  3.35819258           Prob > F      =  0.0001
    Residual |  3.33743545    23  .145105889           R-squared     =  0.5016
-------------+------------------------------           Adj R-squared =  0.4799
       Total |  6.69562803    24  .278984501           Root MSE      =  .38093

------------------------------------------------------------------------------
    fylltemp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wNAO |    -.17356    .037567    -4.62   0.000    -.2512733   -.0958468
       _cons |   1.703436   .1153695    14.77   0.000     1.464776    1.942096
-------------+----------------------------------------------------------------
         rho |   .2951576
------------------------------------------------------------------------------
Durbin-Watson statistic (original)    1.344998
Durbin-Watson statistic (transformed) 1.789412
```

prais 是一种较老的方法，比 **arima** 更加专门化。它的标准误以回归为基础，并假定 ρ 是已知的、而不是估计的。因为这个假定并不真实，由 **prais** 给出的标准误、统计检验和置信区间都不够保险，特别是在小样本中。**prais** 提供了 Durbin-Watson 统计量($d=1.789$)。在这个例子中，Durbin-Watson 检验表明，在拟合这个模型以后没有留下显著的一阶自相关。

14 编程入门

在第 2 章和第 3 章曾经提到过，我们能够通过在文本(ASCII)文件中写入任意序列的 Stata 命令来创建一个简单的程序。Stata 的 Do 文件编辑器(点击 **Window-Do-file Editor** 或点击图标)提供了便利的方式来做这个工作。将 do 文件存盘以后，我们进入 Stata 并键入命令 **do *filename*** 以告诉 Stata 来读文件 *filename.do*，并执行其中所包含的所有命令。使用 Stata 的内置编程语言来编制更成熟和复杂的程序也是可能的。前面各章用过的许多命令实际上都涉及在 Stata 中编写的程序。这些程序中有些源于 Stata 公司，有些则是由 Stata 用户自己编写的，因为他们想完成的特定任务超过了 Stata 的内置性能。

Stata 程序可以访问所有现存的 Stata 性能，可以访问其他程序、而被访问的程序中又可以依次再访问其他程序，还可以使用模型拟合工具包括矩阵代数和最大似然估计。不管我们的目的是大到一种新的统计技术还是小到一项专门的工作，比如，管理某一特定数据，我们编写 Stata 程序的能力都能大大扩展我们能实际完成的工作范围。

关于 Stata 编程有丰富的文献(如《Stata 编程参考手册》(*Stata Programming Reference Manual*)、《Mata 参考手册》(*Mata Reference Manual*)、《用 Stata 做最大似然估计》(*Maximum Likelihood Estimation with Stata*))。这一迷人的题目也包括定期网络教程(见www.stata.com)的内容，同时也在《用户指南》中占据了一个部分。本章仅介绍一些基本的工具，并提供一些应用的示范例子。

基本的概念与工具

我们从一些基本的概念与工具着手，同时伴以前面各章所描述过的 Stata 的功能来作介绍。

do 文件

do 文件是 ASCII(文本)文件，可以用 Stata 的 do 文件编辑器、文字处理器或其他任何文本编辑器来创建。它们的特别之处是保存文件的扩展名为“*.do*”。这个文件能够包含序列的合法 Stata 命令。在 Stata 中，键入以下命令将使 Stata 读取文件 *filename.do* 并且执行其中包含的命令：

```
. do filename
```

filename.do 中的每一条命令，包括最后一条，都必须结束于一个硬回车，除非我

们专门通过命令 **#delimit** 将默认分隔符(delimiter)重新设置为其他字符。比如:

```
#delimit;
```

将设置英文分号";"作为行尾分隔符,因而 Stata 会直到遇到一个分号才认为一行结束。设置分号为分隔符允许一个命令的长度扩展成多个物理行。随后,我们可以再设"回车"(carriage return)为通常的行尾分隔符,使用命令:

```
#delimit cr
```

ado 文件

ado(代表 automatic do)文件也是 ASCII 文件,包含序列 Stata 命令,很像 do 文件。它们的区别是,在运行一个 ado 文件时,我们用不着键入命令 **do *filename***。比如,我们键入命令:

```
. clear
```

与遇到任何其他命令一样,Stata 读到这个命令时先检查是否存在这样命名的内在命令。如果在基础 Stata 可执行命令中并不存在 **clear** 这样一个命令(实际上它并不存在),那么 Stata 下一步就会在其通常的"ado"各目录中去搜寻一个名为 *clear.ado* 的文件。要是 Stata 找到了这样一个文件(它应该会找到),就会执行这个文件中的所有命令。ado 文件的扩展名为"*.ado*"。用户所写的程序通常存放在名为"C:\ado\personal"的目录中,然而大量 Stata 官方的 ado 文件则是安装在"C:\stata\ado"目录中。键入 **sysdir** 就可以看到 Stata 当前所用目录的清单。键入 **help sysdir** 或 **help adopath** 以咨询有关改变它们的建议。

命令 **which** 可揭示某一个命令是否真的是内在的、固定编码的 Stata 命令,或者是定义为 ado 文件了。如果它是一 ado 文件,那么它在哪里存放。比如,**logit** 是一个内置命令[18],但是 **logistic** 命令却是一个 ado 文件,名为 *logistic.ado*:

```
. which logit
built-in command:  logit

. which logistic
C:\STATA\ado\base\l\logistic.ado
*! version 3.1.9  01oct2002
```

这种区别对于大多数用户而言没有多大差别,因为执行 **logit** 和 **logistic** 命令都同样容易,调用它们时用的是类似的命令语法。

程序

do 文件和 ado 文件都可以视为程序的类型,但是 Stata 所使用的"程序"(program)一词则是狭义的,它是指存放于内存并通过键入特定程序名称便可运行的一套命令。do 文件、ado 文件或交互式键入的命令都定义了这样的程序。这种定义开始于声明一个程序名称。比如,要创建一个名为 *count5* 的程序,我们要先写

```
program count5
```

然后应该是实际定义这一程序的那些行。最后,我们要给出 **end** 命令,并随之以硬回车:

```
end
```

---

[18]【译注:**logit** 现在已经是 ado 程序。】

一旦 Stata 读到这些按程序定义的命令,它将会在内存中保留这个程序的定义,而且每当我们将这个程序名作为命令键入时都会运行它:

```
. count5
```

程序实际上使得 Stata 中有了新的命令可用,因而大多数用户用不着知道某一个命令到底是 Stata 本身的,还是一个 ado 文件定义的程序。

当我们开始写一个新程序时,经常建立的是不完全的或是不成功的初步版本。这时命令 **program drop** 提供了基本帮助,让我们从内存中清除程序,以便我们再定义一个新的版本。例如,要从内存中清除程序 *count5*,就键入

```
. program drop count5
```

要是想从内存清除所有的程序(但是不包括数据),就键入

```
. program drop _all
```

局部宏

宏(macro)就是能代表字符串、程序定义结果、以及用户定义值的名称(长度不超过 31 个字符)。而局部宏(local macro)只有在程序之内有所定义时才存在,并且不能从其他程序中进行调用。要建立一个名为 iterate 的局部宏来代表数字 0,就键入

```
        local iterate = 0
```

要想查阅一个局部宏的内容(本例即数字 0),将这个宏的名称置于左单引号和右单引号之内。例如:

```
        display 'iterate'
0
```

于是,要将 iterate 的值加 1,我们要写

```
        local iterate = 'iterate'+1
```

全局宏

全局宏(global macro)与局部宏类似,但是一旦被定义,它们就会留在内存、而且可以被其他程序使用。要查阅一个全局宏的内容,我们要在宏的名称之前加上美元符号"$"(而不是像局部宏那样括以左、右单引号):

```
        global distance = 73
        display $distance * 2
146
```

版本

Stata 的功能和特色已经经历了多年的变化。作为结果,为较早版本(version)的 Stata 所写的程序也许并不能在当前版本中直接运行。命令 version 就是针对这个问题的,以便让原来的程序还能使用。一旦我们告诉 Stata 这个程序是为哪一版写的,Stata 就会作必要的调整使原来的程序能够在新版 Stata 中运行。比如,如果我们在程序的开始写了以下声明,Stata 将按 Stata6 的方式来翻译这个程序的所有命令:

```
        version 6
```

注释

Stata 不会将任何以星号开始的一行作为命令来运行。所以,这样的行可用于在程序中加入注释(comments),或者在一段 Stata 工作期间加入一种交互提示。比如,下一行注释在星号之后说明“这一整行都是注释”:

```
*This entire line is a comment.
```

另外,我们也可以在可执行的命令行中包括一个注释。最简单的方式就是在两个斜线号“ //”之后加上注释(两个斜线之前至少要有一个空格(space))。比如:

```
summarize income education    //this part is the comment
```

加三个斜线(之前也至少要有一个空格)则表明其后直至行尾的内容是一个注释,但是其随后的一条物理行的内容是前面命令的继续,应该被运行。比如:

```
summarize income education    ///this part is the comment
occupation age
```

这个命令的执行结果将会与以下命令的结果相同:

```
summarize income education occupation age
```

不管有没有注释,三个斜线还为程序中的很长命令提供了一种便利方式。比如,下面的两行将会被作为一条 table 命令,尽管它们之间还有硬回车分隔。

```
table gender kids school if contam==1, contents (mean lived///
    median lived count lived)
```

然而,如果我们的程序中有很多特长的命令,那么 #delimit ;命令可能在写和读的时候更为方便(前面有所描述,另可参见 **help delimit**)。

将注释加在一个命令行的中间也是可能的,我们只需要将注释内容用“/* ”和“*/”括起来即可。比如:

```
summarize income /* this is the comment */education occupation
```

如果一行以“ /* ”结尾,而下一行以“*/”开始,那么 Stata 将会略过这个行中断,将两行作为一条命令。这也是在程序中有时能见到的处理特长命令的一个窍门。

循环

有好几种方式可以建立程序的循环(loop)。一个简单方法是使用 forvalues 命令。比如,以下程序做计数从 1 到 5 的循环显示:

```
*Program that counts from one to five
program count5
    version 8.0
    forvalues i=1/5 {
        display `i'
    }
end
```

通过键入这些命令,我们定义了程序 count5。此外,我们也可以用 do 文件编辑器将同样的序列命令存成名为 *count5.do* 的 ASCII 文件。然后键入以下命令让 Stata 来读这个文件:

```
. do count5
```

不论哪一种方式,通过定义程序 count5 我们使其成为一个新的可用命令:

```
. count5
1
2
3
4
5
```

其中的命令

```
forvalues i =1/5 {
```

赋值局部宏 i 从 1 到 5 依次取联贯整数值。另一个命令

```
display `i'
```

要求显示这一宏的内容。其中,宏名称 i 可以任意指定。那么另一个稍有不同的指令便能使我们做从 0 到 100 并按步长 5 来计数(即依次赋值为 0,5,10,…,100):

```
forvalues j =0(5)100 {
```

每一步的赋值不需要非得是整数。要想从 4 到 5 按增量 0.01 来计数(比如,4.00,4.01,4.02,…,5.00),可以将命令写为:

```
forvalues k =4(.01)5 {
```

在开、关大括号{}之间任何有效的 Stata 命令都将被反复执行,对应着每一个赋值。注意:在这行命令中的开括号后什么都没有,但是关括号需要自己单成一行。

命令 foreach 采用的是不同的方法来作循环。不是靠指定一套联贯数值,而是根据我们所给的一个分项清单对应每一项便重复一次。这些分项可以是变量、文件、字符串或者数值。键入 **help foreach** 可参见此命令的语法。

forvalues 和 foreach 建立的循环都是按事先指定的次数来重复。要是我们想让循环一直继续到某些其他条件被满足才停止,那么 while 命令便有用了。按以下一般形式所写的一段程序将反复执行大括号内的命令,只要表达式(expression)被评价为"真":

```
while expression {
    command A
    command B
    . . . .
    }
command Z
```

正如前面的例子一样,关括号"}"并不是处于最后一个命令行的末尾,而是应该自己单独占一行。

当表达式被评价为"假"时,这一循环便结束了,而且 Stata 将继续执行命令 *Z*(*command Z*)。与我们前一个例子类似,这里也是一个使用 while 循环从屏幕显示 iterate 从 1 变到 6 的简单程序:

```
*Program that counts from one to six
  program count6
  version 8.0
  local iterate = 1
  while `iterate'< =6 {
    display `iterate'
```

```
        local iterate = `iterate'+1
        }
    end
```

第二个使用 while 循环的例子出现在 *gossip.ado* 程序中,我们将在本章的后面加以描述。《编程参考手册》中还包括了关于编程循环的更多内容。

如果……否则

if 与 else(*如果与否则*)命令告诉程序,如果一个表达式为真时就做一件事,而在表达式为假时则做另外的事。它们的语法设置如下:

```
if expression {
    command A
    command B
    . . . .
}
else {
    command Z
}
```

比如,下面的一段程序检查局部宏 span 是否为奇数,并且将结果通知用户。

```
if int(`span'/2) != (`span' - 1)/2 {
    display "span is NOT an odd number"
}
else {
    display "span IS an odd number"
}
```

变　元

程序定义新的命令。在某些情况下(如前面的例子 count5),我们想要我们的命令每次被调用时都能做完全相同的事。然而,我们也经常需要一个命令能由变元(argument)加以修改,变元可以是变量名或者是某种选项。我们有两种方法告诉 Stata 如何读取和理解包含变元的命令行。最简单的就是 args 命令。

下面的 do 文件(*listres1.do*)定义了一个程序来做两个变量的回归,然后列出残差绝对值最大的观测案例。

```
* Perform simple regression and list observations with #
* largest absolute residuals.
*   listres1 Yvariable Xvariable # IDvariable
program listres1, sortpreserve
    version 8.0
    args Yvar Xvar number id
    quietly regress `Yvar' `Xvar'
    capture drop Yhat
    capture drop Resid
    capture drop Absres
    quietly predict Yhat
    quietly predict Resid, resid
    quietly gen Absres = abs(Resid)
    gsort -Absres
    drop Absres
    list `id' `Yvar' Yhat Resid in 1/`number'
end
```

其中“args Yvar Xvar number id ”这一行告诉Stata,命令 listres1 应该后接4个变元。这些变元可以是数字、变量名称或者其他由空隔分开的字符串。于是,第一个变元就用了名为 Yvar 的局部宏的内容,第二个变元用了名为 Xvar 的局部宏,如此等等。然后,程序在其他命令中应用这些宏的内容,比如,回归

```
quietly regress `Yvar' `Xvar'
```

这个程序计算残差绝对值(*Absres*),然后用 gsort 命令(后接一个负号置于变量名之前)按从大到小来排序,缺失值排在最后:

```
gsort -Absres
```

命令行中的选项 sortpreserve 使得这个程序“排序稳定化”(sort-stable):即每次所有计算一结束,程序就将数据恢复为原来的顺序。

数据 *nations.dta*,在前面第8章曾经看过,包含109个国家的数据,变量有预期寿命(*life*)、每天人均卡路里(*food*)以及国家名称(*country*)。我们能够打开这个文件,并用它来示范我们的新程序。用 **do** 命令来运行 do 文件 *listres1.do*,从而定义程序 listres1:

```
. do listres1.do
```

然后,我们用新定义的 listres1 命令,并后接其4个变元。第一个变元确定了 *y* 变量,第二个变元确定了 *x* 变量,第三个变元指定了要列出的观测数,第四变元则提供案例识别码。在这个例子中,我们的命令要求的是5条有最大残差绝对值的案例清单。

```
. listres1 life food 5 country

     +------------------------------------------+
     | country    life       Yhat       Resid   |
     |------------------------------------------|
  1. |   Libya      60    76.6901   -16.69011   |
  2. |  Bhutan      44   60.49577   -16.49577   |
  3. |  Panama      72   58.13118    13.86882   |
  4. |  Malawi      45   58.58232   -13.58232   |
  5. | Ecuador      66   52.45305    13.54695   |
     +------------------------------------------+
```

利比亚(Libya)、不丹(Bhutan)和马拉维(Malawi)的实际预期寿命低于按食品供应所做的预测值。相反,巴拿马(Panama)和厄瓜多尔(Ecuador)的预期寿命却高于预测值。

语 法

命令 syntax(语法)提供了更复杂然而也更有用的方式来读一个命令行。下面的 do 文件名为 *listres2.do*,它与我们前一个例子类似,但是它没有用 args,而是用了 syntax:

```
* Perform simple or multiple regression and list
* observations with # largest absolute residuals.
*  listres2 yvar xvarlist [if] [in], number(#) [id(varname)]
program listres2, sortpreserve
version 8.0
syntax varlist(min=1) [if] [in], Number(integer) [Id(string)]
   marksample touse
   quietly regress `varlist' if `touse'
   capture drop Yhat
```

```
    capture drop Resid
    capture drop Absres
    quietly predict Yhat if `touse'
    quietly predict Resid if `touse', resid
    quietly gen Absres = abs(Resid)
    gsort -Absres
    drop Absres
    list `id' `1' Yhat Resid in 1/`number'
end
```

listres2 的目的与以前的 listres1 的目的相同:它要完成回归,然后列出那些有最大残差绝对值的案例来。但是,这个新版本包含着应用 syntax 命令而达到的几个方面的改进。它不再限于像 listres1 那样的两个变量的回归。listres2 可以做任何自变量数目的回归,也包括不设自变量的回归(在这种情况下,预测值等于 y 的平均数,并且残差就是距平均数的离差)。listres2 还可以允许 if 和 in 这类的选项。在 listres2 中用来识别观测案例的备选变量,而在 listres1 中这种识别变量是必须的。比如,我们可以将预期寿命(*life*)对食品(*food*)和能源(*energy*)做回归,然而限制我们的分析只对那些人均 GNP 在 500 美元以上的国家来做。

```
. do listres2.do
. listres2 life food energy if gnpcap > 500, n(6) i(country)

     +-----------------------------------------+
     |  country   life        Yhat       Resid |
     |-----------------------------------------|
  1. | YemenPDR     46    61.34964   -15.34964 |
  2. |  YemenAR     45    59.85839   -14.85839 |
  3. |    Libya     60    73.62516   -13.62516 |
  4. | S_Africa     55     67.9146    -12.9146 |
  5. | HongKong     76    64.64022    11.35978 |
     |-----------------------------------------|
  6. |   Panama     72    61.77788    10.22212 |
     +-----------------------------------------+
```

本例中含 syntax 的行示范了这个命令的一些总的特征:

```
syntax varlist(min=1) [if] [in], Number(integer) [Id(string)]
```

对 listres2 的命令的变量清单中要求至少包括一个变量名在内(varlist(min = 1))。方括号标志了备选的变元,在本例中即为 if 和 in 选择条件以及 id( )选项。对选项中的第一个字母做大写则代表着可以采用最小缩小形式。由于本例含 syntax 的行指定 Number(integer) Id(string),对应的实际命令可以写为:

```
. listres2 life food, number(6) id(country)
```

或者也可写为:

```
. listres2 life food, n(6) i(country)
```

要求局部宏 number 的内容必须是一个整数,而 id 则是一个字符串(比如,一个变量名 *country*)。

这个例子还示范了 marksample 命令,它给子样本(满足了 if 和 in 选择条件的)做标志,以便用于随后的分析。

《编程手册》中对 syntax 命令本身的语法作了简要说明。在实验和学习其他程序时也有助于获得关于这个命令的把握。

程序示范:移动自相关

上一节提供了基本的思路和示范性短程序。在这一节中,我们将这些思路应用于稍微长一些的分析过程新程序。这个过程根据 Topliss(2001)的海洋大气数据来获取时间序列的移动自相关。下面的 do 文件 *gossip.do* 定义了一个称为 gossip 的新命令。那些以星号开始的行或由双斜线注明的部分则注释了这一程序正在做什么。有缩进的行对程序运行并没有影响,但是可以方便程序员的阅读。

```
capture program drop gossip    // FOR WRITING & DEBUGGING; DELETE LATER
program gossip
version 8.0
* Syntax requires user to specify two variables (Yvar and TIMEvar), and
* the span of the moving window.  Optionally, the user can ask to generate


* a new variable holding autocorrelations, to draw a graph, or both.
syntax varlist(min=1 max=2 numeric), SPan(integer) [GENerate(string) GRaph]
if int(`span'/2) != (`span' - 1)/2 {
   display as error "Span must be an odd integer"
}
else {
* The first variable in `varlist' becomes Yvar, the second TIMEvar.
   tokenize `varlist'
       local Yvar `1'
       local TIMEvar `2'
   tempvar NEWVAR
   quietly gen `NEWVAR' = .
   local miss = 0
* spanlo and spanhi are local macros holding the observation number at the
* low and high ends of a particular window.  spanmid holds the observation
* number at the center of this window.
   local spanlo = 0
   local spanhi = `span'
   local spanmid = int(`span'/2)
   while `spanlo' <=   _N -`span' {
       local spanhi = `span' + `spanlo'
       local spanlo = `spanlo' + 1
       local spanmid = `spanmid' + 1
* The next lines check whether missing values exist within the window.
* If they do exist, then no autocorrelation is calculated and we
* move on to the next window.  Users are informed that this occurred.
       quietly summ `Yvar' in `spanlo'/`spanhi'
       if r(N) != `span' {
           local miss = 1
       }
* The value of NEWVAR in observation `spanmid' is set equal to the first
* row, first column (1,1) element of the row vector of autocorrelations
* r(AC) saved by corrgram.
       else {
           quietly corrgram `Yvar' in `spanlo'/`spanhi', lag(1)
           quietly replace `NEWVAR' = el(r(AC),1,1) in `spanmid'
       }
   }
   if "`graph'" != "" {
* The following graph command illustrates the use of comments to cause
* Stata to skip over line breaks, so it reads the next two lines as if
* they were one.
       graph twoway spike `NEWVAR' `TIMEvar', yline(0) ///
           ytitle("First-order autocorrelations of `Yvar' (span `span')")
   }
   if `miss' == 1 {
       display as error "Caution:  missing values exist"
   }
   if "`generate'" != "" {
       rename `NEWVAR' `generate'
       label variable `generate' ///
           "First-order autocorrelations of `Yvar' (span `span')"
   }
}
end
```

正如注释的描述，gossip 要求时间序列(tsset)数据。根据现有时间序列变量，gossip 计算出第二个时间序列变量，它是根据观测的移动窗口(moving window)(比如，按 9 年跨距移动)所计算的时滞 1 的自相关系数。数据 *nao.dta* 包含了北大西洋气候的时间序列，可以用来进行示范：

```
Contains data from C:\data\nao.dta
  obs:           159                       North Atlantic Oscillation &
                                             mean air temperature at
                                             Stykkisholmur, Iceland
 vars:             5                       1 Aug 2005 10:50
 size:         3 498  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
year            int    %ty                    Year
wNAO            float  %9.0g                  Winter NAO
wNAO4           float  %9.0g                  Winter NAO smoothed
temp            float  %9.0g                  Mean air temperature (C)
temp4           float  %9.0g                  Mean air temperature smoothed
-------------------------------------------------------------------------------
Sorted by:  year
```

变量 *temp* 记录了 1841—1999 年期间冰岛西部斯蒂基斯霍尔米的年平均气温。*temp4* 为 *temp* 的修匀值(参见第 13 章)。图 14.1 画出了这两个时间序列。为了区别原始变量 *temp* 和修匀变量 *temp4*，我们对前者用很细的连线，选项为 **clwidth (vthin)**，对后者则用粗连线，选项为 **clwidth (thick)**。键入 **help linewidthstyle** 咨询其他线宽选项。

```
. graph twoway line temp year, clpattern(solid) clwidth(vthin)
      ||  line temp4 year, clpattern(solid) clwidth(thick)
      ||  , ytitle("Temperature, degrees C") legend(off)
```

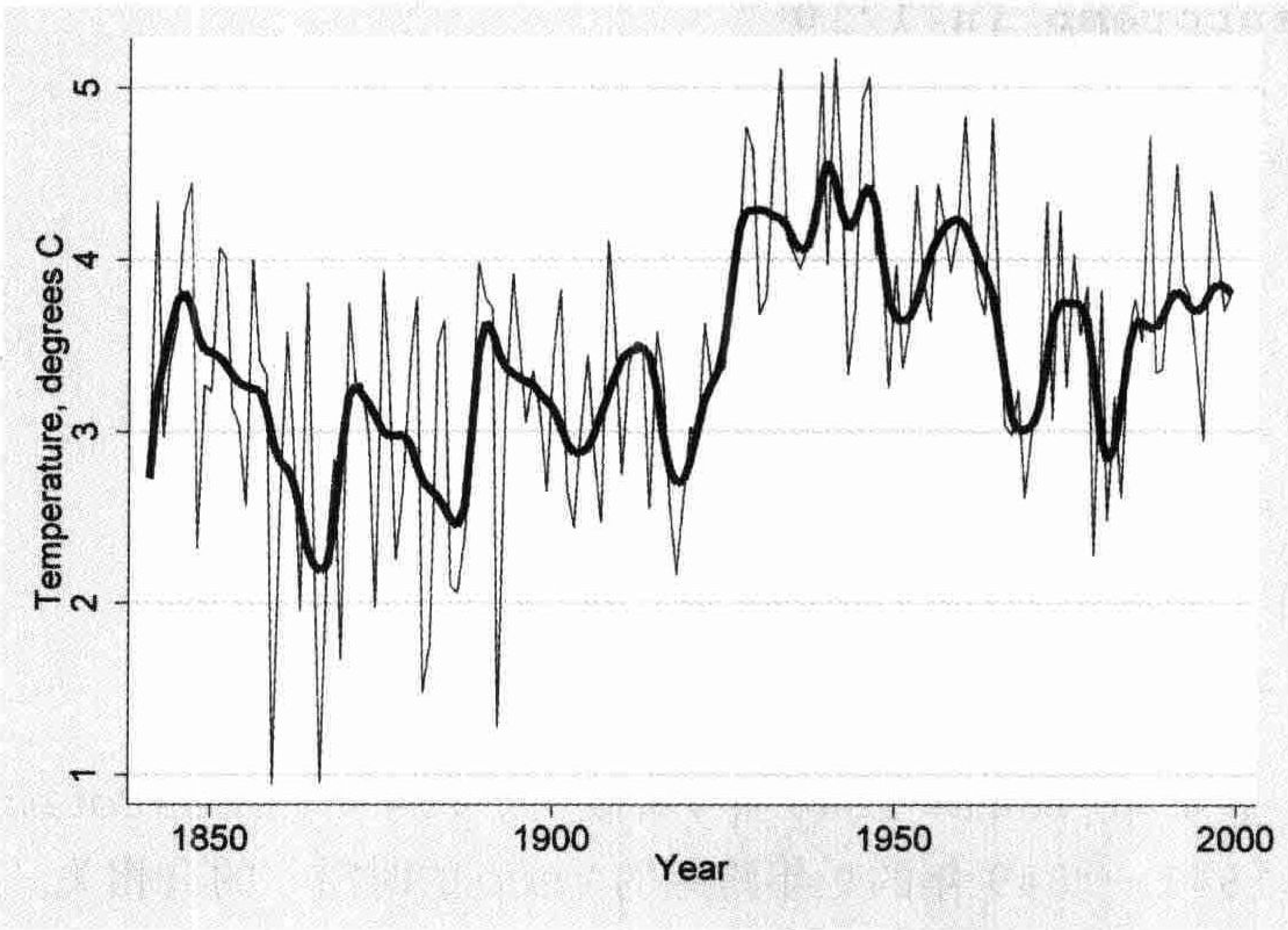


图 14.1

为了在 9 年的移动窗口内计算并画出 *temp* 的序列自相关，我们键入以下命令。它们得到了图 14.2。

```
. do gossip.do
. gossip temp year, span(9) generate(autotemp) graph.
```

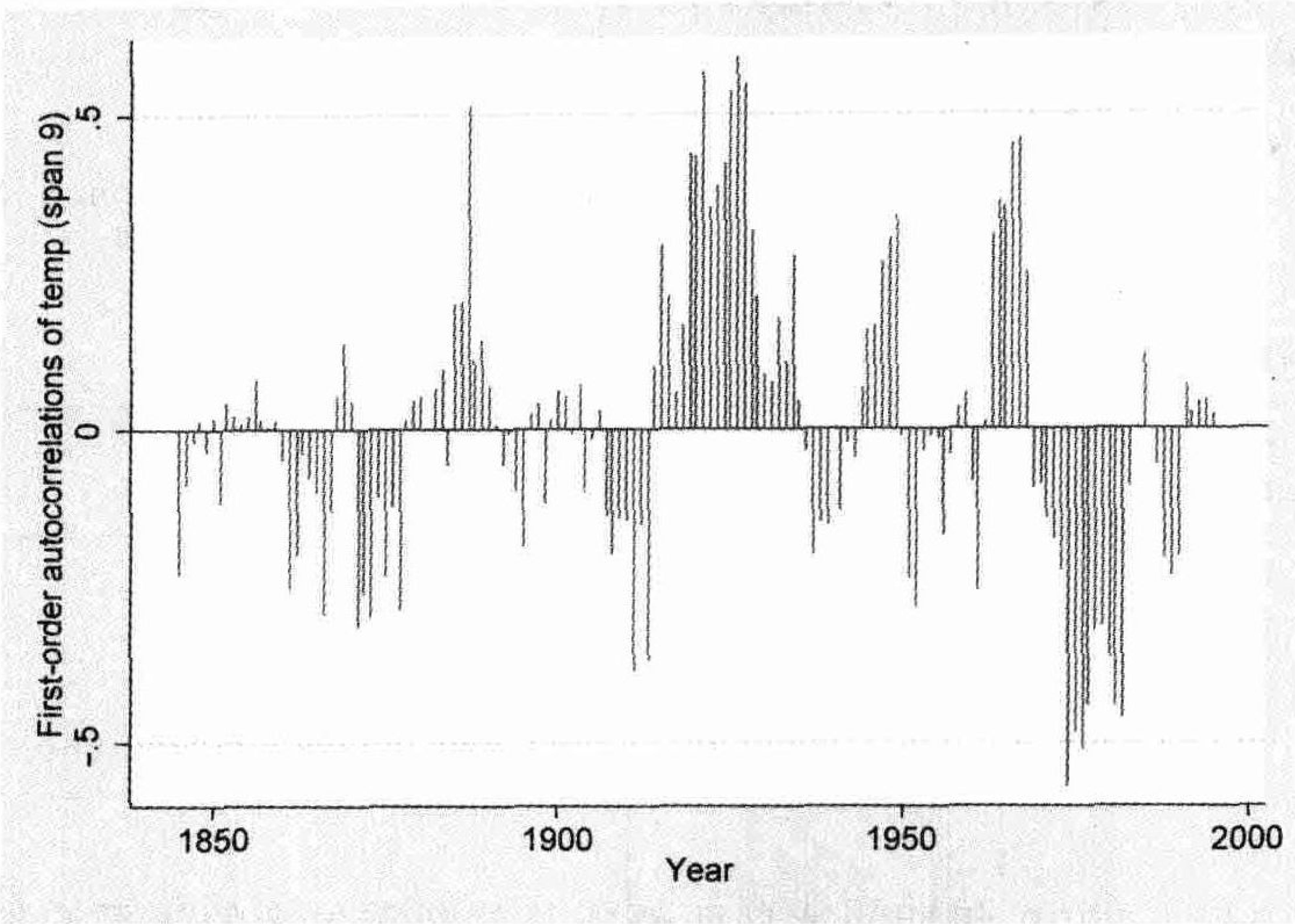


图14.2

除了画出图14.2以外，gossip 还创建了一个名为 *autotemp* 的新变量：

```
. describe autotemp

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
autotemp        float  %9.0g                  First-order autocorrelations of
                                                temp (span 9)

. list year temp autotemp in 1/10

     +--------------------------+
     | year   temp     autotemp |
     |--------------------------|
  1. | 1841   2.73            . |
  2. | 1842   4.34            . |
  3. | 1843   2.97            . |
  4. | 1844   3.41            . |
  5. | 1845   3.62    -.2324837 |
     |--------------------------|
  6. | 1846   4.28    -.0883512 |
  7. | 1847   4.45    -.0194607 |
  8. | 1848   2.32     .0175247 |
  9. | 1849   3.27      -.03303 |
 10. | 1850   3.23     .0181154 |
     +--------------------------+
```

autotemp 值在第一个4年(1841—1844年)时缺失。1845年的 *autotemp* 值(-0.232 483 7)等于从1841—1849年的9年跨距的 *temp* 的时滞1的自相关。这个系数值与我们将键入的以下命令的结果相同：

```
. corrgram temp in 1/9, lag(1)

                                          -1       0       1 -1       0       1
LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1      -0.2325  -0.2398  .66885  0.4135          -|                   -|
```

在1846年，*autotemp* 的值(-0.088 351 2)等于从1842—1850年的9年跨距的

temp 的时滞 1 的自相关。*autotemp* 值在数据的最后 4 年(1996—1999)与最前面的 4 年一样又为缺失。

1920 年代北极显著在变暖,这可以从图 14.1 中的温度曲线看出,在图 14.2 中则表现为一段时期中一致的正自相关。而 1960 年代一段很短暂的正的自相关则对应着气候在变冷。Topliss(2001)曾建议,这种自相关可以作为海洋大气系统变化反馈的指示器。

do 文件 *gossip.do* 是通过几次扩充写出来的,一开始先着手输入部分,诸如命令声明和跨距宏,通过运行这个 do 文件来检查它是否工作,然后再添加其他的部分。并不是所有的试验运行都能得到满意结果。键入以下命令将导致 Stata 将逐行地显示正在运行的程序,所以我们能够确切地看到错误是在哪里发生的:

```
. set trace on
```

以后,我们还能够用以下命令来关闭这一功能:

```
. set trace off
```

gossip.do 中的第一行命令, capture program drop gossip,是要求在再次定义前先将这个程序从内存中清除。这一点在编写和调试程序的阶段是很有帮助的,因为我们以前版本的程序可能尚未完成,或者存在错误。然而,这样的命令行应该在程序已经成熟后予以删除。下一节将描述更多的步骤来使 gossip 成为一个正规的 Stata 命令。

ado 文件

一旦我们相信我们的 do 文件定义的是我们还想再用的程序,我们便可以创建一个 ado 文件使其成为像其他 Stata 命令一样的命令。对于前一个例子 *gossip.do* 而言,这一变化涉及两个步骤:

①用 do 文件编辑器整个删去程序的第一行,它包含在编写和调试阶段用的“DELETE LATER”。我们还可以删去那些注释行。这么做也就删了一些有用的信息,但是将使程序更加紧凑并易于阅读。

②在一个新目录下保存修改后的文件,并改用 .ado 扩展名(比如,*gossip.ado*)。推荐位置为 C:\ado\personal ,如果这个目录和子目录尚不存在,那么你需要自己来创建它们。置于其他位置也不是不可以,但是请先阅读一下《用户手册》中“Stata 从哪里寻找 ado 文件?”这一节。

一旦这些都完成了,我们就能将 gossip 作为 Stata 中正规命令来使用了。以下列出 *gossip.ado* 文件内容的清单:

```
*!  version 2.0
*!  L. Hamilton, Statistics with Stata (2004)
program gossip
version 8.0
syntax varlist(min=1 max=2 numeric), SPan(integer) [GENerate(string) GRaph]
if int(`span'/2) != (`span' - 1)/2 {
    display as error "Span must be an odd integer"
}
else {
    tokenize `varlist'
        local Yvar `1'
        local TIMEvar `2'
```

```
    tempvar NEWVAR
    quietly gen `NEWVAR' = .
    local miss = 0
    local spanlo = 0
    local spanhi = `span'
    local spanmid = int(`span'/2)
    while `spanlo' <= _N -`span' {
        local spanhi = `span' + `spanlo'
        local spanlo = `spanlo' + 1
        local spanmid = `spanmid' + 1
        quietly summ `Yvar' in `spanlo'/`spanhi'
        if r(N) != `span' {
            local miss = 1
        }
        else {
            quietly corrgram `Yvar' in `spanlo'/`spanhi', lag(1)
            quietly replace `NEWVAR' = el(r(AC),1,1) in `spanmid'
        }
    }
    if "`graph'" != "" {
        graph twoway spike `NEWVAR' `TIMEvar', yline(0) ///
            ytitle("First-order autocorrelations of `Yvar' (span `span')")
    }
    if `miss' == 1 {
        display as error "Caution:  missing values exist"
    }
    if "`generate'" != "" {
        rename `NEWVAR' `generate'
        label variable `generate' ///
            "First-order autocorrelations of `Yvar' (span `span')"
    }
}
end
```

这个程序还能进一步加工以使其更灵活、更雅致、更加用户友好化。注意在程序中所加入的前两行都是由"*!"引导的关于来源和版本的注释行。这里注释所说的版本是指 *gossip.ado* 的第 2.0 版,而不是指 Stata 版本(*gossip.ado* 的更早版本曾经出现在本书以前的版本中)。适用这一程序的 Stata 版本在几行之后用 version 命令指定为第 8.0 版。尽管用"*!"来引导注释并不影响程序的运行,但它们在执行 **which** 命令时是能看见的。

```
. which gossip
c:\ado\personal\gossip.ado
*!  version 2.0
*!  L. Hamilton, Statistics with Stata (2004)
```

一旦 *gossip.ado* 被存于 C:\ado\personal 目录中,命令 gossip 就能被随时使用了。如果我们跟着本章做了所有的步骤,那么以前就会定义过 gossip 的初步版,那么在运行新的 ado 文件版本之前,我们应该键入命令来清除内存中的旧的定义:

```
. program drop gossip
```

现在,我们就准备来运行最终的 ado 文件版本了。比如,要是我们想看看数据 *nao.dta* 中变量 *wNAO* 按跨距 15 的自相关,那么我们只需要简单地先打开数据 *nao.dta* 然后键入:

```
. gossip wNAO year, span(15) graph
```

帮助文件

帮助文件(help file)是使用 Stata 时不可或缺的一个方面。对一个像 *gossip.ado* 这样的用户撰写程序,帮助文件就变得更为重要,因为在印刷手册中根本没有相应文件。我们能够为 *gossip.ado* 写一个帮助文件,就是用 Stata 的 do 文件编辑器创建一个名为 *gossip.hlp* 的文本文件。这一帮助文件应该存放于与 *gossip.ado* 相同的 ado 文件的目录里(比如,C:\ado\personal)。

当我们键入 ***help filename*** 时,存在 Stata 公认的 ado 文件目录里的以 *filename.hlp* 形式命名的任何文本文件的内容都将被 Stata 显示在屏幕上。比如,我们可以在编辑器里写入下面的内容,然后将其在目录 C:\ado\personal 中存为 *gossip1 .hlp*。那么,任何时候键入 **help gossip1**,都会导致 Stata 显示这个文本。

```
help for gossip                L. Hamilton

Moving first-order autocorrelations

gossip yvar timevar, span(#) [ generate(newvar) graph ]

Description

calculates first-order autocorrelations of time series
yvar, within a moving window of span #.  For example, if we
specify span(7) gen(new), then the first
through 3rd values of new are missing.  The 4th value of new
equals the lag-1 autocorrelation of yvar across observations 1
through 7.  The 5th value of new equals the lag-1 autocorrelation
of yvar across observations 2 through 8, and so forth.  The last
3 values of new are missing.  See Topliss (2001) for a rationale
and applications of this statistic to atmosphere-ocean data.
Statistics with Stata (2004) discusses the gossip program itself.

gossip requires tsset data.  timevar is the time
variable to be used for graphing.

Options

span(#)   specifies the width of the window for
      calculating autocorrelations.  This option is required;
      # should be an odd integer.

gen(newvar)  creates a new variable holding the
      autocorrelation coefficients.

graph     requests a spike plot of lag-1 autocorrelations vs.
      timevar.

Examples

    . gossip water month, span(13) graph
    . gossip water month, span(9) gen(autowater)
    . gossip water month, span(17) gen(autowater) graph

References

Hamilton, Lawrence C.  2004.  Statistics with Stata.  Pacific Grove,
 CA:  Duxbury.

Topliss, Brenda J.  2001.  "Climate variability I:  A conceptual approach to
ocean-atmosphere feedback."  In Abstracts for AGU Chapman Conference, The
North Atlantic Oscillation, Nov. 28 - Dec 1, 2000, Ourense, Spain.
```

较好的帮助文件包含链接、文本的格式、对话框以及其他特色,这些都可以用 Stata

的标注和控制语言(Stata Markup and Control Language, SMCL)来设计。所有正式的 Stata 帮助文件,如同日志文件以及屏幕显示结果一样,都采用 SMCL 形式。以下是 gossip 的帮助文件的 SMCL 版本。这个文件应被存于目录 C:\ado\personal 中并命名为 *gossip.hlp*,那么键入 **help *gossip***,都会导致可读性更强的正式外观的显示。

```
{smcl}
{* 1aug2003}{...}
{hline}
help for {hi:gossip}{right:(L. Hamilton)}
{hline}

{title:Moving first-order autocorrelations}

{p 8 12}{cmd:gossip} {it:yvar timevar} {cmd:,} {cmdab:sp:an}{cmd:(}
{it:#}{cmd:)} [ {cmdab:gen:erate}{cmd:(}{it:newvar}{cmd:)}
{cmdab:gr:aph} ]


{title:Description}

{p}{cmd:gossip} calculates first-order autocorrelations of time series
{it:yvar}, within a moving window of span {it:#}.  For example, if we
specify {cmd:span(}7{cmd:)} {cmd:gen(}{it:new}{cmd:)}, then the first
through 3rd values of {it:new} are missing.  The 4th value of {it:new}
equals the lag-1 autocorrelation of {it:yvar} across observations 1
through 7.  The 5th value of {it:new} equals the lag-1 autocorrelation
of {it:yvar} across observations 2 through 8, and so forth.  The last
3 values of {it:new} are missing.  See Topliss (2001) for a rationale
and applications of this statistic to atmosphere-ocean data.
{browse "http://www.stata.com/bookstore/sws.html":Statistics with Stata}
 (2004) discusses the {cmd:gossip} program itself.{p_end}

{p}{cmd:gossip} requires {cmd:tsset} data.  {it:timevar} is the time
variable to be used for graphing.{p_end}


{title:Options}

{p 0 4}{cmd:span(}{it:#}{cmd:)} specifies the width of the window for
calculating autocorrelations.  This option is required; {it:#} should be
 an odd integer.
{p 0 4}{cmd:gen(}{it:newvar}{cmd:)} creates a new variable holding the
autocorrelation coefficients.

{p 0 4}{cmd:graph} requests a spike plot of lag-1 autocorrelations vs.
{it:timevar}.


{title:Examples}

{p 8 12}{inp:. gossip water month, span(13) graph}{p_end}
{p 8 12}{inp:. gossip water month, span(9) gen(autowater)}{p_end}
{p 8 12}{inp:. gossip water month, span(17) gen(autowater) graph}{p_end}


{title:References}

{p 0 4}Hamilton, Lawrence C.  2004.
{browse "http://www.stata.com/bookstore/sws.html":Statistics with Stata}.
 Pacific Grove, CA:  Duxbury.{p_end}

{p 0 4}Topliss, Brenda J.  2001.  "Climate variability I:  A conceptual
approach to ocean-atmosphere feedback."  In Abstracts for AGU Chapman
Conference, The North Atlantic Oscillation, Nov. 28 - Dec 1, 2000, Ourense,
Spain. citation.{p_end}
```

这个帮助文件开始于 {smcl},它告诉 Stata 按 SMCL 格式来处理这个文件。大括号 { } 括住了 SMCL 代码,它们就属于形式 {command: text} 或 {command argument:text}。下面的例子说明这些代码是如何被解释的:

{hline}	画一条水平线。
{hi:gossip}	高亮显示文本"gossip"。
{title:Moving...}	将文本"Moving..."作为标题显示。
{right:L. Hamilton}	将文本"L. Hamilton" 右对齐。
{p 8 12}	将后面的文本作为一个段落,其中第一行缩进 8 列,随后各行缩进 12 列。
{cmd:gossip}	将文本"gossip"作为一个命令显示。即,用当前为命令专门定义的颜色和字形、字号来显示"gossip"。
{it:yvar}	将文本"yvar"显示为斜体字形(itlics)。
{cmdab:sp:an}	将文本"span"作为一个命令显示,其中字母"sp"标注为最小缩写[19]。
{p}	将后面的文本作为一个段落,直到见 {p_end} 结束。
{browse "http://www.stata.com/bookstore/sws.html":Statistics...}	链接文本"Statistics with Stata"到互联网址(URL) http://www.stata.com/bookstore/sws.html。点击文本"Statistics with Stata"即可将用户的浏览器链接到这一网址。

《编程手册》中提供了使用这些及其他 SMCL 命令的详细说明。

矩阵代数

矩阵代数(matrix algebra)提供了统计建模的基础工具。Stata 的矩阵命令和矩阵编程语言(Mata)太丰富多样,不可能在这里充分描述,有关题目需要参阅它自己的参考手册(《Mata 参考手册》),此外还可以参阅《编程参考手册》和《用户指南》中的很多篇幅。有关 Mata 语言以及 Stata 第 9 版的新变化的信息,请咨询以上资料来源。本节用一些例子来示范以前的矩阵命令,但它们现在仍能工作(因而在每个程序开始时要加入一个 **version 8.0** 命令)。

Stata 内置命令 **regress** 执行常规最小二乘法(OLS)回归,自然还有其他工作。但是为了练习,我们可以自己来写一个 OLS 程序。后面的 *ols1.do* 定义了一个简单的回归程序,它除了计算和显示回归系数估计向量以外不做其他的事。以下是很熟悉的 OLS 公式以及这个程序的文本:

$$b = (X'X)^{-1}X'y$$

---

[19]【译注:显示为加粗,最小缩写指只需要健入 **sp** 即可代表命令 **span**。】

```
* A very simple program, "ols1" estimates linear regression
* coefficients using ordinary least squares (OLS).
program ols1
   version 8.0
* The syntax allows only for a variable list with one or more
* numeric variables.
   syntax varlist(min=1 numeric)
* "tempname..." assigns names to temporary matrices to be used in this
* program.  When ols1 has finished, these matrices will be dropped.
   tempname crossYX crossX crossY b
* "matrix accum..." forms a cross-product matrix.  The K variables in
* varlist, and the N observations with nonmissing values on all K variables,
* comprise an N row, K column data matrix we might call yX.
* The cross product matrix crossYX equals the transpose of yX times yX.
* Written algebraically:
*          crossYX = (yX)'yX
   quietly matrix accum `crossYX' = `varlist'
* Matrix crossX extracts rows 2 through K, and columns 2 through K,
* from crossYX:
*          crossX = X'X
   matrix `crossX' = `crossYX'[2...,2...]
* Column vector crossY extracts rows 2 through K, and column 1 from crossYX:
*          crossY = X'y
   matrix `crossY' = `crossYX'[2...,1]
* The column vector b contains OLS regression coefficients, obtained by
* the classic estimating equation:
*          b = inverse(X'X)X'y
   matrix `b' = syminv(`crossX') * `crossY'
* Finally, we list the coefficient estimates, which are the contents of b.
   matrix list `b'
end
```

在以上 *ols1.do* 程序中的每一条命令都有注释来说明。从没有注释行的程序版本 *ols2.do*(后面)中可以更清楚地看到矩阵命令：

```
program ols2
   version 8.0
   syntax varlist(min=1 numeric)
   tempname crossYX crossX crossY b
   quietly matrix accum `crossYX' = `varlist'
   matrix `crossX' = `crossYX'[2...,2...]
   matrix `crossY' = `crossYX'[2...,1]
   matrix `b' = syminv(`crossX') * `crossY'
   matrix list `b'
end
```

在 *ols1.do* 和 *ols2.do* 中都没有用 **in** 或 **if** 选择条件，都没有语法错误，也没有命令选项。它们也没有计算通常与回归配套的标准误、置信区间或其他辅助统计量。为了看看这些命令到底在做什么，我们将分析一个关于核电厂的小数据(*reactor.dta*)：

```
Contains data from c:\data\reactor.dta
  obs:             5                          Reactor decommissioning costs
                                                (from Brown et al. 1986)
 vars:             6                          1 Aug 2005 10:50
 size:           130 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
site            str14  %14s                   Reactor site
decom           byte   %8.0g                  Decommissioning cost, millions
capacity        int    %8.0g                  Generating capacity, megawatts
years           byte   %9.0g                  Years in operation
start           int    %8.0g                  Year operations started
close           int    %8.0g                  Year operations closed
-------------------------------------------------------------------------------
Sorted by:  start
```

关闭一座反应堆的成本随其发电量和运行年数而提高，正如用 **regress** 的回归结果所示：

```
. regress decom capacity years

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  2,     2) =  189.42
       Model |  4666.16571     2  2333.08286           Prob > F      =  0.0053
    Residual |  24.6342883     2  12.3171442           R-squared     =  0.9947
-------------+------------------------------           Adj R-squared =  0.9895
       Total |     4690.80     4     1172.70           Root MSE      =  3.5096

------------------------------------------------------------------------------
       decom |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    capacity |   .1758739   .0247774     7.10   0.019      .0692653    .2824825
       years |   3.899314   .2643087    14.75   0.005      2.762085    5.036543
       _cons |  -11.39963   4.330311    -2.63   0.119     -30.03146     7.23219
------------------------------------------------------------------------------
```

我们自制的程序 *ols2.do* 也得到了同样的回归系数：

```
. do ols2.do

. ols2 decom capacity years

__000003[3,1]
                decom
capacity    .1758739
   years   3.8993139
   _cons  -11.399633
```

尽管它的结果是正确的，这个最简单的 OLS2 程序缺乏许多在我们在建模时所需的许多特色功能。下面的 ado 文件 *ols3.ado* 定义了一个名为 ols3 的改进程序。这个程序允许用 **in** 或 **if** 选择条件，并且允许随意指定置信区间的水平。它计算出回归系数，并且采用表格形式工整地显示回归系数及其标准误、t 检验、置信区间。

```
*! version 2.0 1aug2003
*! Matrix demonstration:  more complete OLS regression program.
program ols3, eclass
   version 8.0
   syntax varlist(min=1 numeric) [in] [if] [, Level(integer $S_level)]
   marksample touse
   tokenize "`varlist'"
   tempname crossYX crossX crossY b hat V
   quietly matrix accum `crossYX' = `varlist' if `touse'
   local nobs = r(N)
   local df = `nobs' - (rowsof(`crossYX') - 1)
   matrix `crossX' = `crossYX'[2...,2...]
   matrix `crossY' = `crossYX'[2...,1]
   matrix `b' = (syminv(`crossX') * `crossY')'
   matrix `hat' = `b' * `crossY'
   matrix `V' = syminv(`crossX') * (`crossYX'[1,1] - `hat'[1,1])/`df'
   ereturn post `b' `V', dof(`df') obs(`nobs') depname(`1') ///
      esample(`touse')
   ereturn local depvar "`1'"
   ereturn local cmd "ols3"
   if `level' < 10 | `level' > 99 {
      display as error "level( ) must be between 10 and 99 inclusive."
      exit 198
   }
   ereturn display, level(`level')
end
```

因为 *ols3.ado* 是个 ado 文件，我们只需要简单键入 ols3 来作为一个命令：

```
. ols3 decom capacity years

------------------------------------------------------------------------------
       decom |      Coef.   Std. Err.       t     P>|t|      [95% Conf. Interval]
-------------+----------------------------------------------------------------
    capacity |   .1758739    .0247774      7.10    0.019      .0692653    .2824825
       years |   3.899314    .2643087     14.75    0.005      2.762085    5.036543
       _cons |  -11.39963    4.330311     -2.63    0.119     -30.03146     7.23219
------------------------------------------------------------------------------
```

ols3.ado 包含许多我们已经熟悉的元素，包括 syntax 和 marksample 命令，还有前面 *ols1.do* 和 *ols2.do* 中已经见过的种种 matrix 运算。注意右单引号(')为"矩阵转置"运算符。我们将系数向量（syminv(\`crossX') * \`crossY')的转置则像以下那样写为：

```
(syminv(`crossX') * `crossY')'
```

ols3 程序被定义为 e 类(e-class)，表示其为统计模型估计(estimation)类的命令：

```
program ols3, eclass
```

e 类程序保存其结果时都带着标注 e( )。在前一 **ols3** 命令运行后，已经有了以下内容：

```
. ereturn list

scalars:
                  e(N) =  5
               e(df_r) =  2

macros:
                e(cmd) : "ols3"
             e(depvar) : "decom"

matrices:
                  e(b) :  1 x 3
                  e(V) :  3 x 3

functions:
             e(sample)

. display e(N)
5

. matrix list e(b)

e(b)[1,3]
      capacity        years        _cons
y1    .1758739    3.8993139   -11.399633

. matrix list e(V)

symmetric e(V)[3,3]
             capacity        years        _cons
capacity    .00061392
   years   -.00216732     .0698591
   _cons   -.01492755     -.942626    18.751591
```

从 e 类程序得到的 e( )结果将在下一个 e 类命令执行之前一直保存在内存中。与此相反，r 类程序比如 **summarize** 保存其结果时会用 r( )标注，并且只保存到下一个 e

类或 r 类命令的执行。

在 *ols3.ado* 中的几个 ereturn 行都是保存这种 e( )结果,并使用它们作输出显示:

```
ereturn post `b' `V', dof(`df') obs(`nobs') depname(`1') ///
     esample(`touse')
```

上述命令设置 e( )结果的内容,包括系数向量(b)和方差协方差矩阵(V)。这使得 **help estimates** 和 **help postest** 所详细介绍的那些后续估计(post-estimation)功能成为可用。此命令的选项定义了残差自由度(df)、估计所用的观测案例数(nobs)、因变量名称(`1 ′表示当我们运行 tokenize varlist(即表示变量表)时取得的第一个宏的内容)以及估计样本的标记(touse)。

```
ereturn local depvar "`1'"
```

这一命令在 tokenize varlist 后设置因变量(即宏 1)为宏 e(depvar)的内容。

```
ereturn local cmd "ols3"
```

这一命令将命令的名称 ols3 设置为宏 e(cmd)的内容。

```
ereturn display, level(`level')
```

命令 ereturn display 显示我们前面的 ereturn post 形成的系数表。这一输出表为标准 Stata 格式:它的前两列为系数估计(从 b 提供)和标准误(即 V 的对角线元素的平方根)。其他列分别为 t 检验(第一列除以第二列)、双尾 t 概率以及根据 ols3 的有关命令行指定的水平(默认水平为 95%)计算的置信区间。

自助法

自助法(bootstrap)指一种不断在现有数据中重复随机重置(with replacement)取样的过程。这里我们不再相信一个估计量的理论抽样分布,而是根据经验方法近似计算出这一分布来。抽取 k 次样本规模为 n 的自助样本(从规模也为 n 的原始数据样本中重置抽取)可以得到 k 个新的估计。这些自助法估计的分布提供了用于估计标准误或置信区间的经验基础(Efron 和 Tibshirani, 1986;有关介绍还可见 Stine 在 Fox 和 Long 主编著作中的论文,1990)。自助法在一些情况下很有吸引力,比如,当统计研究遇到理论上难以解决的问题,或者当常规理论的假定条件站不住脚的场合。

与蒙特卡罗模拟(Monte Carlo simulation)需要制造数据不同,自助法完全是根据实际数据来进行工作。作为示范,我们将使用数据 *islands.dta*,其中包含 8 个太平洋群岛的地区面积和生物多样性的指标(引自 Cox 和 Moore,1973)。

```
Contains data from c:\data\islands.dta
  obs:             8                          Pacific island biodiversity
                                                (Cox & Moore 1993)
 vars:             4                          1 Aug 2005 10:50
 size:           208 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
island          str15  %15s                   Island group
area            float  %9.0g                  Land area, km^2
birds           byte   %8.0g                  Number of bird genera
plants          int    %8.0g                  Number flowering plant genera
-------------------------------------------------------------------------------
Sorted by:
```

设想现在我们要为鸟类种属的平均数计算置信区间。通常对平均数的置信区间是从正态分布假定推导出来。然而，当其为一个偏态分布时我们可能并不想做这种假定，甚至就这样极小的样本($n=8$)也几乎导致我们拒绝正态假定：

```
. sktest birds

                   Skewness/Kurtosis tests for Normality
                                                      ------- joint ------
    Variable |  Pr(Skewness)   Pr(Kurtosis)  adj chi2(2)     Prob>chi2
-------------+-------------------------------------------------------
       birds |      0.079          0.181           4.75         0.0928
```

自助法提供了一种更为经验性的方法来形成置信区间。一个 r 类命令 **summarize, detail** 悄悄地将其结果暂存了一系列的宏。这些宏中的一些为：

r(N)	观测案例数
r(mean)	平均数
r(skewness)	偏度
r(min)	最小值
r(max)	最大值
r(p50)	第 50 分位数，即中位数
r(Var)	方差
r(sum)	合计
r(sd)	标准差

暂存这些结果简化了自助法形成任何统计量的工作。要想基于 1 000 次重复取样为鸟类平均数取得自助置信区间，并且将结果存为一个新文件 *boot1.dta*，可键入以下命令(其输出包含一个关于缺失值潜在问题的警告提示(Warning)，但是它并不适用于这些数据)：

```
. bs "summarize birds, detail"  "r(mean)", rep(1000) saving(boot1)

command:      summarize birds , detail
statistic:    _bs_1       = r(mean)

Warning:  Since summarize is not an estimation command or does not set
          e(sample), bootstrap has no way to determine which observations are
          used in calculating the statistics and so assumes that all
          observations are used.  This means no observations will be excluded
          from the resampling due to missing values or other reasons.

          If the assumption is not true, press Break, save the data, and drop
          the observations that are to be excluded.  Be sure the dataset in
          memory contains only the relevant data.
```

```
Bootstrap statistics                              Number of obs    =         8
                                                  Replications     =      1000

------------------------------------------------------------------------------
Variable      | Reps  Observed       Bias  Std. Err. [95% Conf. Interval]
--------------+---------------------------------------------------------------
        _bs_1 | 1000    47.625  -.475875  12.39088   23.30986   71.94014   (N)
              |                                        25.75    74.8125   (P)
              |                                           27      78.25  (BC)
------------------------------------------------------------------------------
Note:  N    = normal
       P    = percentile
       BC   = bias-corrected
```

命令 **bs** 的声明括在双引号中(**"summarize *birds*, detail"**),表明要自助估计的是什么分析。在这其后是指定什么统计量要作自助估计,同样也是被括在自己的双引号中(**"r(mean)"**)。还可以列入更多的统计量,但是在每种统计量之间用一个空格分开。上述例子还指定另外两个选项:

rep(1000) 要求重复 1 000 次,即抽取 1 000 个自助样本。

saving(*boot1*) 将这 1 000 个自助样本平均数存入一个名为 *boot1 .dta* 的数据文件。

命令 **bs** 的输出表显示了所完成的重复次数和要自助估计的统计量的"观测的"(Observed,即原始样本的)值。在本例中,*birds* 的平均数值为 47.625。表中还显示了估计偏差(Bias)、标准误(Stn. Err.)以及三种类型的置信区间([95% Conf. Interval])。在这里,"偏差"指所要统计量的 k 个自助样本值的平均数(比如,1 000 个自助样本的 *birds* 平均数的平均数)减去观测的统计值。估计的标准误等于 k 个自助统计量值的标准差(比如,1 000 个自助样本的 *birds* 平均数的标准差)。这一自助标准误(12.39)小于用命令 **ci** 计算出的常规标准误(13 .38):

```
. ci birds

    Variable |       Obs          Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
       birds |         8        47.625    13.38034        15.98552    79.26448
```

命令 **bs** 的输出表中按正态近似(N)的置信区间按以下公式计算:

$$观测的样本统计量 \pm t \times 自助标准误$$

其中 t 是从自由度为 $k-1$ 的理论 t 分布中所选的概率度值。通常的建议是当自助分布表现为无偏并近似于正态分布时再使用这种置信区间。

输出表中的百分位数(P)置信区间就是简单使用自助分布的相应百分位数(95% 置信区间对应着第 2.5 和第 97.5 这两个百分位点)作为区间的上下限。当自助分布表现为无偏但是呈非正态时,这种置信区间可能比较恰当。

偏差修正(BC,即 bias-corrected)置信区间也使用了自助分布的百分位数,但是在选择百分位数时则是按自助值小于或等于观测统计量的比例所调整的正态分布中取得的。当存在很大偏差(一个经验规则是当偏差超过 1 个标准误的 25%)时,这种置信区间可能更可取。

因为我们将自助结果存入了名为 *boot1 .dta* 的文件,所以如果需要,我们便可以将其取出并更仔细地检查自助分布。选项 **saving(*boot1*)** 创建一个有 1 000 观测的数据,其中名为 *_bs_1* 的变量就是每一个自助样本的平均数。

```
Contains data from c:\data\boot1.dta
  obs:         1 000                          bs: summarize birds, detail
 vars:             1                          1 Aug 2005 15:10
 size:         8 000  (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
_bs_1           float  %9.0g                  r(mean)
-------------------------------------------------------------------------------
Sorted by:
```

```
. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       _bs_1 |      1000    47.14912    12.39088     14.625       92.5
```

注意这 1 000 个自助平均数的标准差就等于前面 bs 输出表中所示的标准误(12.39)。这 1 000 个自助平均数的平均数减去观测(即原始数据)平均数就等于偏差:

$$47.149\,12 - 47.625 = -0.475\,88$$

图 14.3 展示了这 1 000 个样本平均数的分布,以及用垂直线标注的原始样本平均数(47.625)。这个分布显示出轻微的正偏态,但是与理论正态分布的差别不大。

```
. histogram _bs_1, norm bcolor(gs10) xaxis(1 2) xline(47.625)
      xlabel(47.635, axis(2)) xtitle("", axis(2))
```

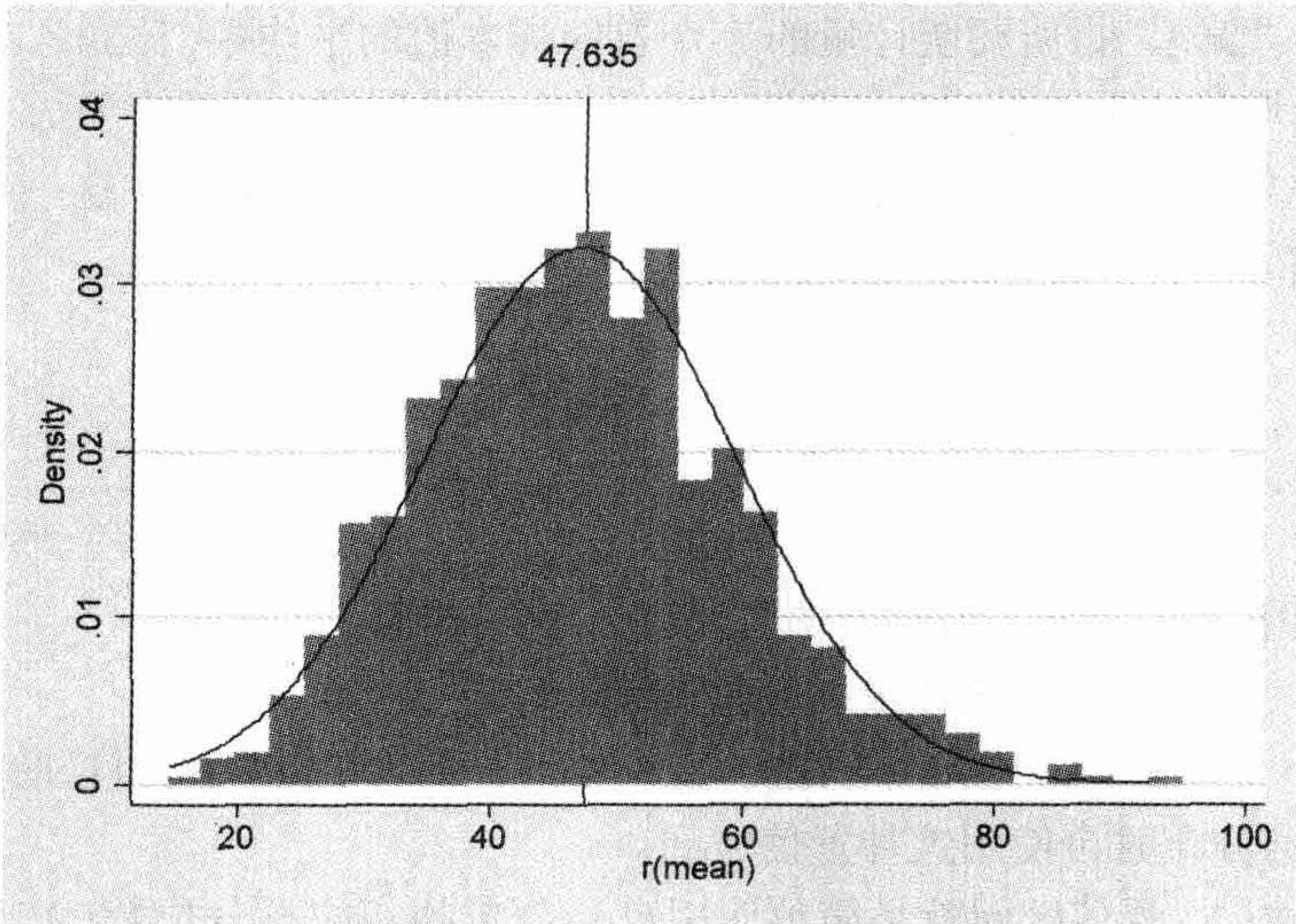


图 14.3

生物学家已经观察到,生物多样性、或植物和生物的不同种类,倾向于随着海岛的面积而增加。在数据集 *islands.dta* 中,我们有鸟类和开花植物的数据来检验这个命题。正如预期的一样,在鸟类品种 *birds* 和面积 *area* 之间存在着很强的线性关系:

```
. regress birds area

      Source |       SS       df       MS              Number of obs =       8
-------------+------------------------------           F(  1,     6) =  162.96
       Model |  9669.83255     1  9669.83255           Prob > F      =  0.0000
    Residual |  356.042449     6  59.3404082           R-squared     =  0.9645
-------------+------------------------------           Adj R-squared =  0.9586
       Total |   10025.875     7  1432.26786           Root MSE      =  7.7033
```

```
------------------------------------------------------------------------------
       birds |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        area |   .0026512   .0002077    12.77   0.000      .002143    .0031594
       _cons |   13.97169    3.79046     3.69   0.010     4.696773    23.24662
------------------------------------------------------------------------------
```

如本章以前所述,e 类命令 **regress** 会保存一套 e( ) 结果。它还创建或更新一套包含模型系数(_b[*varname*])和标准误(_se[*varname*])的系统变量。要对前一个回归的斜率以及 *y* 截距应用自助法,并将结果存于文件 *boot2.dta*,就键入:

```
. bs "regress birds area" "_b[area] _b[_cons]", rep(1000)
     saving(boot2)

command:      regress birds area
statistics:   _bs_1      = _b[area]
              _bs_2      = _b[_cons]

Bootstrap statistics                              Number of obs    =         8
                                                  Replications     =      1000

------------------------------------------------------------------------------
Variable     |  Reps  Observed       Bias  Std. Err.  [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _bs_1 |  1000  .0026512 -.0000737  .0003345    .0019947   .0033077   (N)
             |                                        .0019759   .0029066   (P)
             |                                          .00199   .0029246  (BC)
       _bs_2 |  1000  13.97169  .6230986  3.637705    6.833275   21.11011   (N)
             |                                        7.891942   21.74494   (P)
             |                                        6.949539   19.73012  (BC)
------------------------------------------------------------------------------
Note:  N   = normal
       P   = percentile
       BC  = bias-corrected
```

area 的系数自助分布严重偏态(偏度 =4.12)。尽管平均数的自助分布(图 14.3)近似为正态,并且得到的自助置信区间窄于理论置信区间,但是对这个回归例子应用自助法却取得了较大的标准误和更宽的置信区间。

就回归而言,**bs** 通常完成所谓的"数据再抽样"(data resampling),换句话说就是在不改变数据的情况下再抽样。有另外一种程序称为"残差再抽样"(residual resampling)(即只对残差再抽样),就需要做一些编程工作了。有另外两个命令可为这样自制的自助法提供便利:

bsample 从现有数据中回置性地抽取一个样本,并取代内存中的数据。

bootstrap 运行用户定义的程序,重复 reps( )次,自助样本规模为 size( )。

《基础参考手册》提供了使用 **bootstrap** 命令的例子及程序。

蒙特卡罗模拟

蒙特卡罗模拟(Monte Carlo simulation)形成并分析人造数据的许多样本,允许研究者研究他们所用统计技术的长期行为。命令 **simulate** 使设计一个模拟轻而易举,只需要再附以少量的编程。本节提供两个示例。

要做一个模拟,我们就需要定义一个程序来创建一套随机数据,对其加以分析,并且将所关心的结果保存在内存里。下面我们来看一个定义为 r 类的程序(能够暂存 r( ) 结果),名为 central。这个程序随机产生标准正态分布的变量 *x* 的 100 个值。然后它再由一个"不纯的正态分布"产生另一个变量 *w* 的 100 个值;这个不纯

(contaminated)的正态分布是由0.95概率的N(0,1)和0.05概率的N(0,10)构成的。不纯正态分布经常被用于稳健研究,以模拟那些含有少数异常误差的变量。对上述所产生的两个变量,central 取得其平均数和中位数。

```
* Creates a sample containing n=100 observations of variables x and w.
* x~N(0,1)                                        x is standard normal
* w~N(0,1) with p=.95, w~N(0,10) with p=.05      w is contaminated normal
* Calculates the mean and median of x and w.
* Stored results:   r(xmean)   r(xmedian)   r(wmean)   r(wmedian)
program central, rclass
   version 8.0
   drop _all
   set obs 100
   generate x = invnorm(uniform())
   summarize x, detail
   return scalar xmean = r(mean)
   return scalar xmedian = r(p50)
   generate w = invnorm(uniform())
   replace w = 10*w if uniform() < .05
   summarize w, detail
   return scalar wmean = r(mean)
   return scalar wmedian = r(p50)
end
```

由于我们定义 **central** 为 r 类命令,就像 **summarize** 一样,它可以将自己的结果保存于 r( ) 宏中。**central** 建立4个这样的宏:变量 *x* 的平均数 r(xmean) 和中位数 r(xmedian),以及变量 *w* 的平均数 r(wmean)和中位数 r(wmedian)。

一旦定义了 **central**,不管是采用 do 文件、或是 ado 文件、或是键入交互命令,我们都能用命令 **simulate** 来调用这个程序。从5 000个随机样本建立包含 *x* 和 *w* 的平均数和中位数的新数据,键入:

```
. simulate "central"  xmean = r(xmean)  xmedian = r(xmedian)
     wmean = r(wmean)  wmedian = r(wmedian), reps(5000)

command:       central
statistics:    xmean      = r(xmean)
               xmedian    = r(xmedian)
               wmean      = r(wmean)
               wmedian    = r(wmedian)
```

这个命令基于 **central** 的每一次重复的 r( ) 结果建立了4个新变量 *xmean*、*xmedian*、*wmean*、*wmedian*。

```
. describe

Contains data
  obs:         5 000                            simulate: central
 vars:             4                            1 Aug 2005 17:50
 size:       100 000 (99.6% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
xmean           float  %9.0g                  r(xmean)
xmedian         float  %9.0g                  r(xmedian)
wmean           float  %9.0g                  r(wmean)
wmedian         float  %9.0g                  r(wmedian)
-------------------------------------------------------------------------------
Sorted by:
```

来自5 000个样本的这些平均数和中位数的平均数都接近于0,与我们关于样本平

```
. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       xmean |      5000   -.0015915    .0987788  -.4112561   .3699467
     xmedian |      5000   -.0015566    .1246915  -.4647848   .4740642
       wmean |      5000   -.0004433    .2470823   -1.11406   .8774976
     wmedian |      5000    .0030762    .1303756  -.4584521   .5152998
```

均数和中位数都能提供 x 和 w 的真实总体平均数(0)的无偏估计的期望是一致的。同样符合理论预测的是,当应用于正态分布的变量时,平均数展示了比中位数具有更小的不同样本之间的变异。*xmedian* 的标准差为 0.125,比 *xmean* 的标准差(0.099)大很多。另一方面,当蒙特卡罗模型应用于有特异值的变量 w 时,便出现了相反的结果:*wmedian* 的标准差却比 wmean 的标准差小得多(0.130 对比 0.247)。这一蒙特卡罗试验展示出,中位数在不纯分布含有特异值的条件下来测量中心趋势时要相对更稳定,然而平均数却不太好,并在不同样本之间有着的更大变异。图 14.4 提供了箱线图的形象化比较(附带地示范一下如何控制箱线图中特异值标注符号):

```
. graph box xmean xmedian wmean wmedian, yline(0) legend(col(4))
      marker(1, msymbol(+)) marker(2, msymbol(Th))
      marker(3, msymbol(Oh)) marker(4, msymbol(Sh))
```

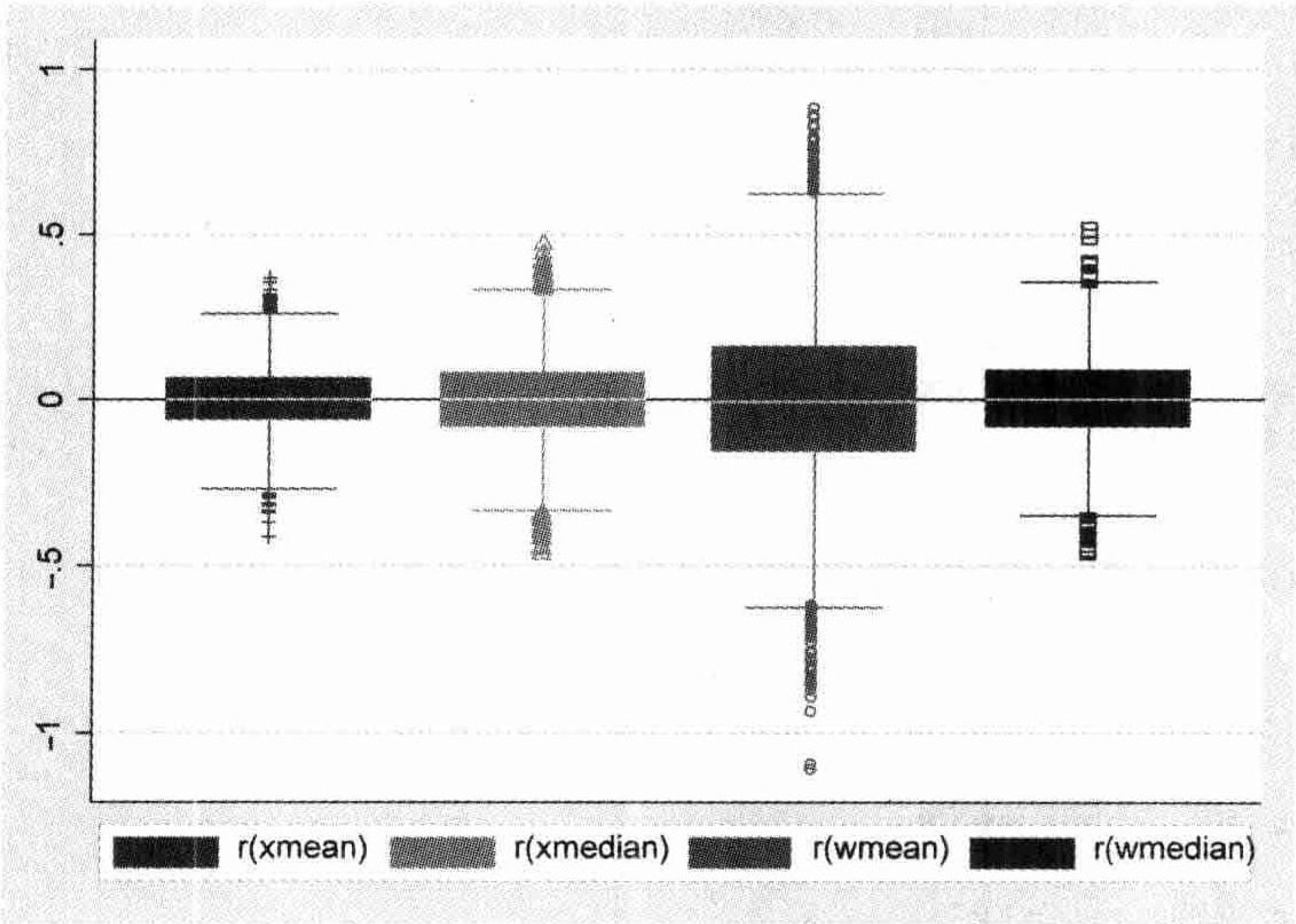


图 14.4

我们最后的例子扩展到稳健方法,并且还联系到本书中几个方法。程序 regsim 将产生 100 个案例的 x 观测(服从标准正态分布)和两个 y 变量。*y1* 是 x 的线性函数再加上标准正态误差,*y2* 也是 x 的线性函数但是再加上不纯正态误差。这些变量允许我们探测在正态误差条件下与非正态误差条件下的不同回归方法有怎样的行为。将应用四种方法:常规最小二乘法(**regress**),稳健回归(**rreg**),百分位数回归(**qreg**);以及百分位回归及自助法标准误(**bsqreg** ,做 500 次重复)。这些方法之间的差别在第 9 章中已经做过讨论。程序 regsim 应用每一种方法做 *y1* 对 x 的回归、然后做 *y2* 对 x 的回归。对于这一练习,程序是由 ado 文件 *regsim.ado* 来定义的,这一文件存放在目录“C:\ado\personal”中。

```
program regsim, rclass
* Performs one iteration of a Monte Carlo simulation comparing
* OLS regression (regress) with robust (rreg) and quantile
* (qreg and bsqreg) regression.  Generates one n = 100 sample
* with x ~ N(0,1) and y variables defined by the models:
*
*   MODEL 1:      y1 = 2x + e1          e1 ~ N(0,1)
*
*   MODEL 2:      y2 = 2x + e2          e2 ~ N(0,1) with p = .95
*                                       e2 ~ N(0,10) with p = .05
*
* Bootstrap standard errors for qreg involve 500 repetitions.
*
   version 8.0
   if "`1'" == "?" {
       #delimit ;
       global S_1 "b1 b1r se1r b1q se1q se1qb
           b2 b2r se2r b2q se2q se2qb";
       #delimit cr
       exit
   }
   drop _all
   set obs 100
   generate x = invnorm(uniform())
   generate e = invnorm(uniform())
   generate y1 = 2*x + e
   reg y1 x
       return scalar B1 = _b[x]
   rreg y1 x, iterate(25)
       return scalar B1R = _b[x]
       return scalar SE1R = _se[x]
   qreg y1 x
       return scalar B1Q = _b[x]
       return scalar SE1Q = _se[x]
   bsqreg y1 x, reps(500)
       return scalar SE1QB = _se[x]
   replace e = 10 * e if uniform() < .05
   generate y2 = 2*x + e
   reg y2 x
       return scalar B2 = _b[x]
   rreg y2 x, iterate(25)
       return scalar B2R = _b[x]
       return scalar SE2R = _se[x]
   qreg y2 x
       return scalar B2Q = _b[x]
       return scalar SE2Q = _se[x]
   bsqreg y2 x, reps(500)
       return scalar SE2QB = _se[x]
end
```

这一 r 类程序保存 8 个回归分析所估计的回归系数或标准误结果。这些结果文件的命名规则如下：

r(B1)	*y1* 对 *x* 的 OLS 回归的系数
r(B1R)	*y1* 对 *x* 的稳健回归的系数
r(SE1R)	模型 1 的稳健回归系数的标准误

如此等等。所有的稳健回归和百分位数回归都涉及多次迭代：对于 **rreg** 一般需要迭代 5 ~10 次，对于 **qreg** 一般需要迭代 5 次，对于要求 500 次自助重复估计的 **bsqreg** 而言，每一次取样的每个估计大约需要迭代 5 次。于是，一次运行 **regsim** 命令需要做

2 000多次回归。以下命令只要求做 5 次迭代[20]。

```
. simulate "regsim" b1 = r(B1) b1r = r(B1R) se1r = r(SE1R)
      b1q = r(B1Q) se1q = r(SE1Q) se1qb = r(SE1QB) b2 = r(B2)
      b2r = r(B2R) se2r = r(SE2R) b2q = r(B2Q) se2q = r(SE2Q)
      se2qb = r(SE2QB), reps(5)
```

一般你应该先试运行像这样一个很小的模拟来试验一下在你电脑上大约要用多长时间。然而要是为了研究的目的,我们将需要大得多的试验。数据集 *regsim.dta* 包含了涉及 5 000 次重复的 regsim 的结果,这一个运行就做了 1 000 万个以上的回归,花了一夜的时间。这一试验取得的回归系数和标准误估计的概要统计如下:

```
. describe

Contains data from C:\data\regsim.dta
  obs:          5 000                          Monte Carlo estimates of b in
                                                 5000 samples of n=100
 vars:             12                          2 Aug 2005 08:17
 size:        260 000 (99.0% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label       variable label
-------------------------------------------------------------------------------
b1              float  %9.0g                   OLS b (normal errors)
b1r             float  %9.0g                   Robust b (normal errors)
se1r            float  %9.0g                   Robust SE[b] (normal errors)
b1q             float  %9.0g                   Quantile b (normal errors)
se1q            float  %9.0g                   Quantile SE[b] (normal errors)
se1qb           float  %9.0g                   Quantile bootstrap SE[b]
                                                 (normal errors)
b2              float  %9.0g                   OLS b (contaminated errors)
b2r             float  %9.0g                   Robust b (contaminated errors)
se2r            float  %9.0g                   Robust SE[b] (contaminated
                                                errors)
b2q             float  %9.0g                   Quantile b (contaminated errors)
se2q            float  %9.0g                   Quantile SE[b] (contaminated
                                                errors)
se2qb           float  %9.0g                   Quantile bootstrap SE[b]
                                                 (contaminated errors)
-------------------------------------------------------------------------------
Sorted by:

. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          b1 |      5000    2.000828     .102018   1.631245   2.404814
         b1r |      5000    2.000989    .1052277   1.603106   2.391946
        se1r |      5000    .1041399    .0109429   .0693786   .1515421
         b1q |      5000    2.001135    .1309186   1.471802   2.536621
        se1q |      5000    .1262578    .0281738   .0532731   .2371508
-------------+--------------------------------------------------------
       se1qb |      5000    .1362755     .032673   .0510808     .29979
          b2 |      5000    2.006001    .2484688   .9001114   3.050552
         b2r |      5000    2.000399    .1092553   1.633241   2.411423
        se2r |      5000    .1081348    .0119274   .0743103   .1560973
         b2q |      5000    2.000701     .137111   1.471802   2.536621
-------------+--------------------------------------------------------
        se2q |      5000    .1328431    .0299644   .0542015   .2594844
       se2qb |      5000    .1436366    .0346679   .0589409   .3006417
```

图 14.5 画出了这些回归系数分布的箱线图。为了使这个箱线图更容易看,我们用了选项 **legend(symxsize(2) colgap(4))**,这个选项设置图例内部符号的宽度,并

[20]【译注:因此试验结果将与本书中结果明显不同!】

且将列间距设为小于原本的默认尺寸。**help legend _ option** 和 **help relativesize** 可以提供更多有关这些选项的信息。

```
. graph box b1 b1r b1q b2 b2r b2q, ytitle("Estimates of slope (b=2)")
      yline(2)
      legend(row(1) symxsize(2) colgap(4)
         label(1 "OLS 1") label(2 "robust 1") label(3 "quantile 1")
         label(4 "OLS 2") label(5 "robust 2") label(6 "quantile 2"))
```

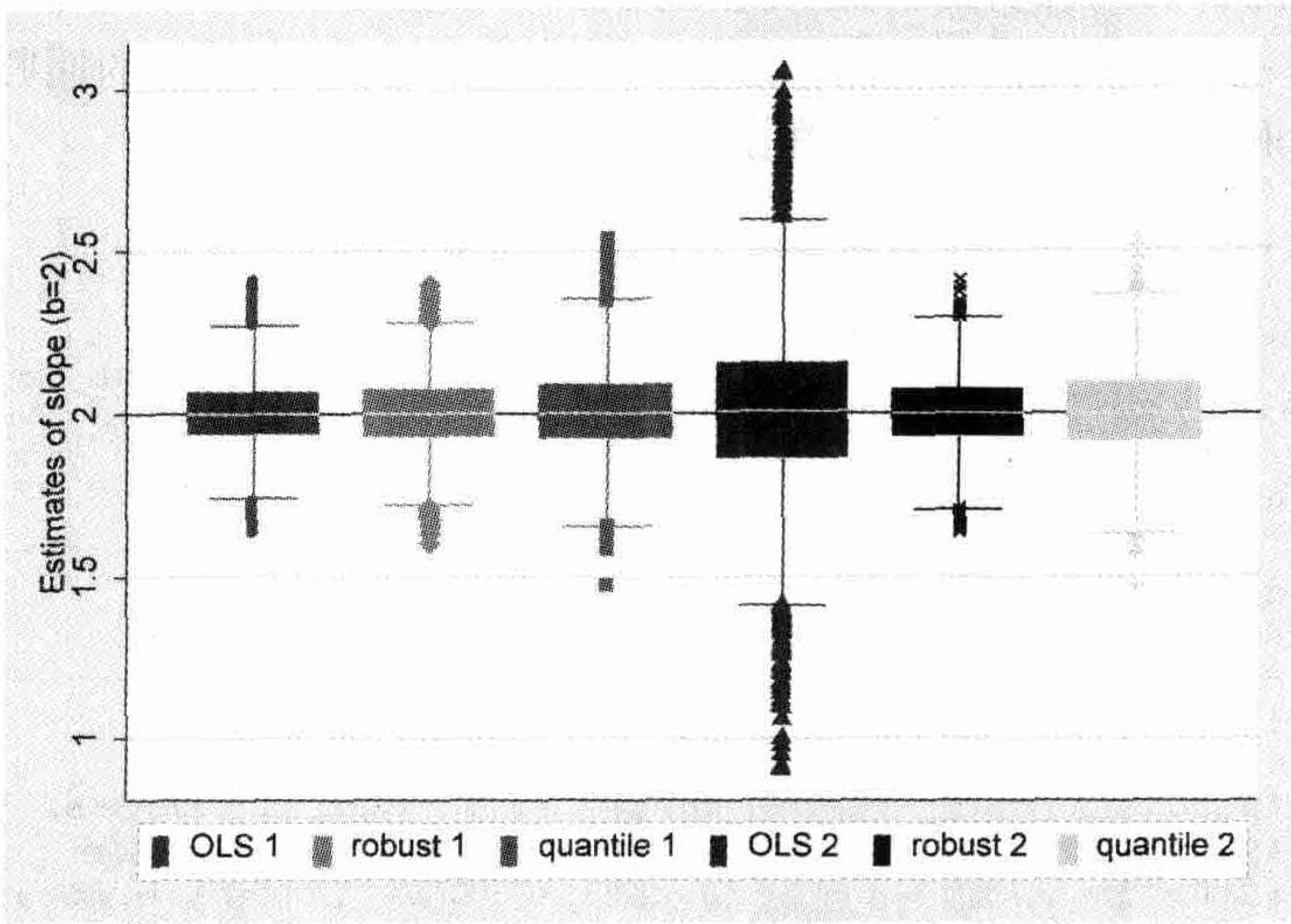


图 14.5

这三种回归方法(OLS、稳健、分位数)对这两种模型的平均系数估计与其真值 β =2 之间的差距并不显著。这能够通过 t 检验加以确认:

```
. ttest b2r = 2

One-sample t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
     b2r |    5000    2.000399    .0015451    .1092553     1.99737    2.003428
------------------------------------------------------------------------------
Degrees of freedom: 4999

                          Ho: mean(b2r) = 2

     Ha: mean < 2              Ha: mean != 2              Ha: mean > 2
       t =   0.2585              t =   0.2585               t =   0.2585
   P < t =   0.6020        P > |t| =   0.7960           P > t =   0.3980
```

于是,所有回归方法都得到了 β 的无偏估计,但是在它们的样本变异或效率性上却有所不同。应用于正态误差模型 1,OLS 估计是最高效的,正如著名的高斯—马尔可夫定理引导我们所期望的那样。OLS 系数的观测标准差为 0.102 0,稳健回归则相应为 0.105 2,而百分位回归的相应统计为 0.130 9。相对效率表达了 OLS 系数的观测方差是其他某种估计方法观测方差的百分比,它提供比较这类统计量的一种标准方式:

```
. quietly summarize b1

. global Varb1 = r(Var)

. quietly summarize b1r

. display 100*($Varb1/r(Var))
93.992612
```

```
. quietly summarize b1q
. display 100*($Varb1/r(Var))
60.722696
```

上述计算使用了由 **summarize** 取得的 r(Var)方差结果。我们先求出 OLS 回归估计 *b1* 的方差,并将其存放于全局宏 Varb*1* 中。接下来依次求出稳健回归估计 *b1 r* 的方差,以及百分位回归估计 *b1 q* 的方差,并且将其与 Varb1 进行比较。结果表明,在应用正态误差模型时,稳健回归的效率为 OLS 估计的 94%,接近于稳健方法理论上应该有的 95% 的大样本效率(Hamilton, 1992a)。相比之下,在正态误差模型时的百分位回归只取得了 61% 的相对效率。

对不纯误差模型作类似的计算却讲出另一番不同的故事。OLS 在正态误差时是最佳(最有效率的)估计,但是应用于不纯误差时,它却成为最差的估计:

```
. quietly summarize b2
. global Varb2 = r(Var)
. quietly summarize b2r
. display 100*($Varb2/r(Var))
517.20057
. quietly summarize b2q
. display 100*($Varb2/r(Var))
328.3971
```

不纯误差模型中的特异值造成 OLS 系数估计在不同样本之间变异很大,正如在图14.5中的第四个箱线图所示。这些 OLS 回归系数的方差已经比相应稳健回归系数的方差大了5 倍以上,比百分位回归系数的方差大了 3 倍以上。换句话说,在有特异值的情况下,已经证明稳健回归和百分位回归要比 OLS 估计稳定得多,可以得到相应较小的标准误和较窄的置信区间。在正态误差和不纯误差两种模型中,稳健回归都胜于百分位回归。

图 14.6 以散点图形式展示了 OLS 估计与稳健估计的 5 000 对回归系数的比较。OLS 系数(纵轴)关于真值 2.0 的变异比 **rreg** 系数(横轴)的相应变异更大。

```
. graph twoway scatter b2 b2r, msymbol(p) ylabel(1(.5)3, grid)
    yline(2) xlabel(1(.5)3, grid) xline(2)
```

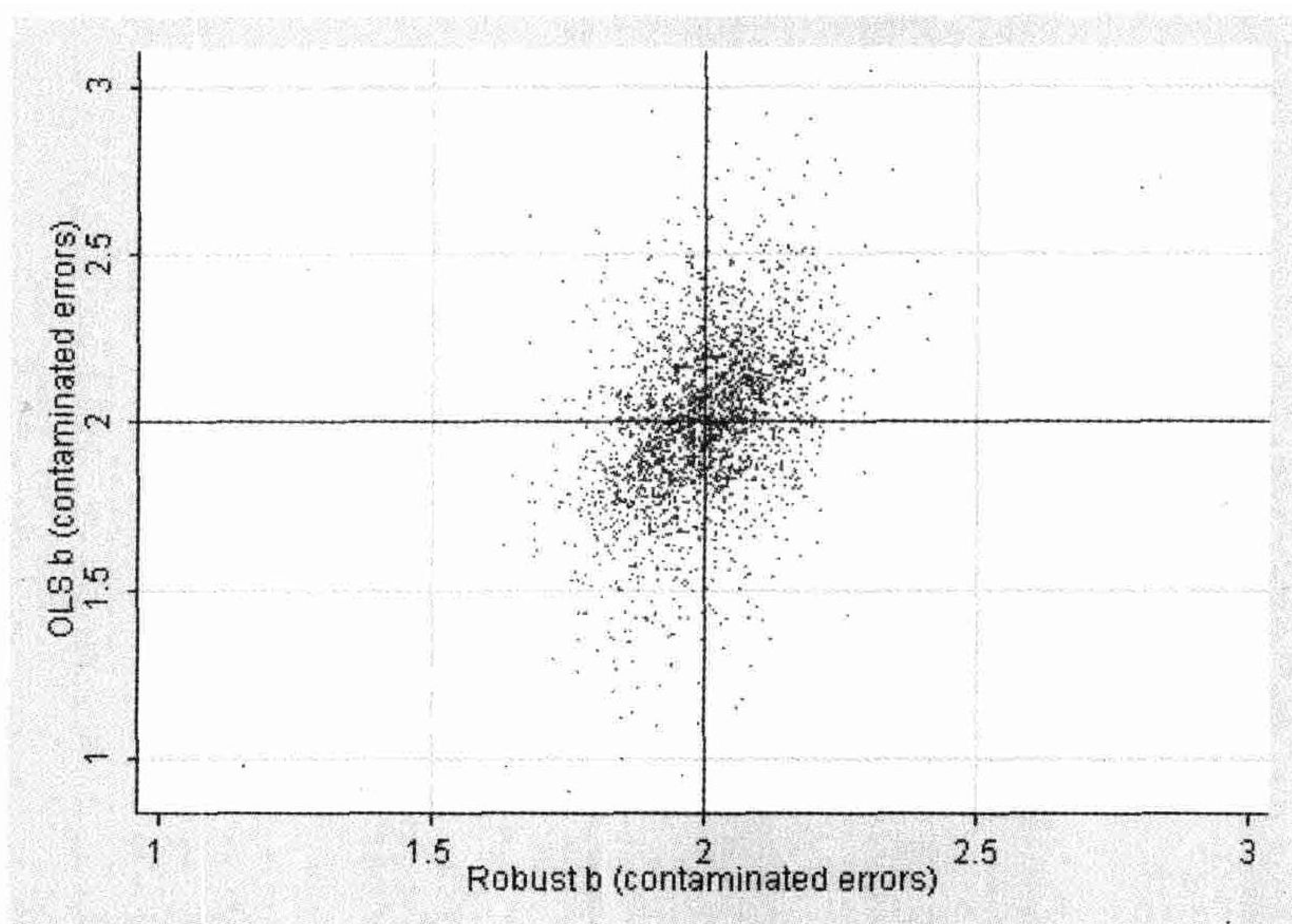


图 14.6

这一试验还提供了各种方法和模型的估计标准误的信息。平均估计标准误与系数的观测标准差不同。对于稳健标准误来说这一差距很小,小于 1%。对于理论推导的百分位回归标准误,这一差距表现得较大一点,在 3% 至 4% 之间。令人最不满意的估计就是由 **bsqreg** 取得的自助法百分位回归估计。自助法标准误的平均数超过了 *b1q* 和 *b2q* 的观测标准差为 4%至 5%。自助法显然高估了样本之间差异。

蒙特卡罗模拟在现代统计研究中已经成为关键的方法,它在统计教学中同样起着越来越大的作用。这些示例展示了使用 Stata 能多么容易地完成蒙特卡罗模拟。

参考文献

Barron's Educational Series. 1992. *Barron's Compact Guide to Colleges*, 8th ed. New York: Barron's Educational Series.

Beatty, J. Kelly, Brian O'Leary and Andrew Chaikin (eds.). 1981. *The New Solar System*. Cambridge, MA: Sky.

Belsley, D. A., E. Kuh and R. E. Welsch. 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. New York: John Wiley & Sons.

Box, G. E. P., G. M. Jenkins and G. C. Reinsel. 1994. *Time Series Analysis: Forecasting and Control*. 3rd ed. Englewood Cliffs, NJ: Prentice-Hall.

Brown, Lester R., William U. Chandler, Christopher Flavin, Cynthia Pollock, Sandra Postel, Linda Starke and Edward C. Wolf. 1986. *State of the World1986*. New York: W. W. Norton.

Buch, E. 2000. *Oceanographic Investigations off West Greenland1999*. Copenhagen: Danish Meteorological Institute.

CDC (Centers for Disease Control). 2003. Web site: http://www.cdc.gov.

Chambers, John M., William S. Cleveland, Beat Kleiner and Paul A. Tukey (eds.). 1983. *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth.

Chatfield, C. 1996. *The Analysis of Time Series: An Introduction*, 5th edition. London: Chapman & Hall.

Chatterjee, S., A. S. Hadi and B. Price. 2000. *Regression Analysis by Example*, 3rd edition. New York: John Wiley & Sons.

Cleveland, William S. 1994. *The Elements of Graphing Data*. Monterey, CA: Wadsworth.

Cleves, Mario, William Gould and Roberto Gutierrez. 2004. *An Introduction to Survival Analysis Using Stata*, revised edition. College Station, TX: Stata Press.

Cook, R. Dennis and Sanford Weisberg. 1982. *Residuals and Influence in Regression*. New York: Chapman & Hall.

Cook, R. Dennis and Sanford Weisberg. 1994. *An Introduction to Regression Graphics*. New York: John Wiley & Sons.

Council on Environmental Quality. 1988. *Environmental Quality* 1987—1988. Washington, DC: Council on Environmental Quality.

Cox, C. Barry and Peter D. Moore. 1993. *Biogeography: An Ecological and Evolutionary Approach*. London: Blackwell Publishers.

Cryer, Jonathan B. and Robert B. Miller. 1994. *Statistics for Business: Data Analysis and Modeling*, 2nd edition. Belmont, CA: Duxbury Press.

Davis, Duane. 2000. *Business Research for Decision Making*, 5th edition. Belmont, CA: Duxbury Press.

DFO (Canadian Department of Fisheries and Oceans). 2003. Web site:
http://www.meds-sdmm.dfo-mpo.gc.ca/alphapro/zmp/climate/IceCoverage_e.shtml.

Diggle, P. J. 1990. *Time Series: A Biostatistical Introduction*. Oxford: Oxford University Press.

Efron, Bradley and R. Tibshirani. 1986. "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy." *Statistical Science* 1(1):54-77.

Enders, W. 2003. *Applied Econometric Time Series*, 2nd edition. New York: John Wiley & Sons.

Everitt, Brian S., Savine Landau and Morven Leese. 2001. *Cluster Analysis*, 4th edition. London: Arnold.

Federal, Provincial, and Territorial Advisory Commission on Population Health. 1996. *Report on the Health of Canadians*. Ottowa: Health Canada Communications.

Fox, John. 1991. *Regression Diagnostics*. Newbury Park, CA: Sage Publications.

Fox, John and J. Scott Long. 1990. *Modern Methods of Data Analysis*. Beverly Hills: Sage Publications.

Frigge, Michael, David C. Hoaglin and Boris Iglewicz. 1989. "Some implementations of the boxplot." *The American Statistician* 43(1):50-54.

Gould, William, Jeffrey Pitblado and William Sribney. 2003. *Maximum Likelihood Estimation with Stata*, 2nd edition. College Station, TX: Stata Press.

Hamilton, Dave C. 2003. "The Effects of Alcohol on Perceived Attractiveness." Senior Thesis. Claremont, CA: Claremont McKenna College.

Hamilton, James D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Hamilton, Lawrence C. 1985a. "Concern about toxic wastes: Three demographic predictors." *Sociological Perspectives* 28(4):463-486.

Hamilton, Lawrence C. 1985b. "Who cares about water pollution? Opinions in a small- town crisis." *Sociological Inquiry* 55(2):170-181.

Hamilton, Lawrence C. 1992a. *Regression with Graphics: A Second Course in Applied Statistics*. Pacific Grove, CA: Brooks/Cole.

Hamilton, Lawrence C. 1992b. "Quartiles, outliers and normality: Some Monte Carlo results." Pp. 92-95 in Joseph Hilbe (ed.) *Stata Technical Bulletin Reprints*, *Volume* 1. College Station, TX: Stata Press.

Hamilton, Lawrence C. 1996. *Data Analysis for Social Scientists*. Belmont, CA: Duxbury Press.

Hamilton, Lawrence C., Benjamin C. Brown and Rasmus Ole Rasmussen. 2003. "Local dimensions of climatic change: West Greenland's cod-to-shrimp transition." *Arctic* 56(3):271-282.

Hamilton, Lawrence C., Richard L. Haedrich and Cynthia M. Duncan. 2003. "Above and below the water: Social/ecological transformation in northwest Newfoundland." *Population and Environment* 25(2): 101-121.

Hamilton, Lawrence C. and Carole L. Seyfrit. 1993. "Town-village contrasts in Alaskan youth aspirations." *Arctic* 46(3):255-263.

Hardin, James and Joseph Hilbe. 2001. *Generalized Linear Models and Extensions*. College Station, TX: Stata Press.

Hoaglin, David C., Frederick Mosteller and John W. Tukey (eds.). 1983. *Understanding Robust and Exploratory Data Analysis*. New York: John Wiley & Sons.

Hoaglin, David C., Frederick Mosteller and John W. Tukey (eds.). 1985. *Exploring Data Tables, Trends and Shape*. New York: John Wiley & Sons.

Hosmer, David W., Jr. and Stanley Lemeshow. 1999. *Applied Survival Analysis*. New York: John Wiley & Sons.

Hosmer, David W., Jr. and Stanley Lemeshow. 2000. *Applied Logistic Regression*, 2nd edition. New York: John Wiley & Sons.

Howell, David C. 1999. *Fundamental Statistics for the Behavioral Sciences*, 4th edition. Belmont, CA: Duxbury Press.

Howell, David C. 2002. *Statistical Methods for Psychology*, 5th edition. Belmont, CA: Duxbury Press.

Iman, Ronald L. 1994. *A Data-Based Approach to Statistics*. Belmont, CA: Duxbury Press.

Jentoft, Svein and Trond Kristoffersen. 1989. "Fishermen's co-management: The case of the Lofoten fishery." *Human Organization* 48(4):355-365.

Johnson, Anne M., Jane Wadsworth, Kaye Wellings, Sally Bradshaw and Julia Field. 1992. "Sexual lifestyles and HIV risk." *Nature* 360(3 December):410-412.

Johnston, Jack and John DiNardo. 1997. *Econometric Methods*, 4th edition. New York: McGraw-Hill.

Keller, Gerald, Brian Warrack and Henry Bartel. 2003. *Statistics for Management and Economics*, abbreviated 6th edition. Belmont, CA: Duxbury Press.

League of Conservation Voters. 1990. *The* 1990 *National Environmental Scorecard*. Washington, DC: League of Conservation Voters.

Lee, Elisa T. 1992. *Statistical Methods for Survival Data Analysis*, 2nd edition. New York: John Wiley & Sons.

Li, Guoying. 1985. "Robust regression." Pp. 281-343 in D. C. Hoaglin, F. Mosteller and J. W. Tukey (eds.) *Exploring Data Tables, Trends and Shape*. New York: John Wiley & Sons.

Long, J. Scott. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage Publications.

Long, J. Scott and Jeremy Freese. 2003. *Regression Models for Categorical Outcomes Using Stata*, revised edition. College Station, TX: Stata Press.

MacKenzie, Donald. 1990. *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*. Cambridge, MA: MIT.

Mallows, C. L.. 1986. "Augmented partial residuals." *Technometrics* 28:313-319.

Mayewski, P. A., G. Holdsworth, M. J. Spencer, S. Whitlow, M. Twickler, M. C. Morrison, K. K. Ferland and L. D. Meeker. 1993. "Ice-core sulfate from three northern hemisphere sites: Source and temperature forcing implications." *Atmospheric Environment* 27A(17/18):2915-2919.

Mayewski, P. A., L. D. Meeker, S. Whitlow, M. S. Twickler, M. C. Morrison, P. Bloomfield, G. C. Bond, R. B. Alley, A. J. Gow, P. M. Grootes, D. A. Meese, M. Ram, K. C. Taylor and W. Wumkes. 1994. "Changes in atmospheric circulation and ocean ice cover over the North Atlantic during the last 41 000 years." *Science* 263:1747-1751.

McCullagh, D. W. Jr. and J. A. Nelder. 1989. *Generalized Linear Models*, 2nd edition. London: Chapman & Hall.

Nash, James and Lawrence Schwartz. 1987. "Computers and the writing process." *Collegiate Microcomputer* 5(1):45-48.

National Center for Education Statistics. 1992. *Digest of Education Statistics* 1992. Washington, DC: U.S.

Government Printing Office.

National Center for Education Statistics. 1993. *Digest of Education Statistics* 1993. Washington, DC: U. S. Government Printing Office.

Newton, H. Joseph and Jane L. Harvill. 1997. *StatConcepts: A Visual Tour of Statistical Ideas*. Pacific Grove, CA: Duxbury Press.

Pagano, Marcello and Kim Gauvreau. 2000. *Principles of Biostatistics*, 2nd edition. Belmont, CA: Duxbury Press.

Rabe-Hesketh, Sophia and Brian Everitt. 2000. *A Handbook of Statistical Analysis Using Stata*, 2nd edition. Boca Raton, FL: Chapman & Hall.

Report of the Presidential Commission on the Space Shuttle Challenger Accident. 1986. Washington, DC.

Rosner, Bernard. 1995. *Fundamentals of Biostatistics*, 4th edition. Belmont, CA: Duxbury Press.

Selvin, Steve. 1995. *Practical Biostatistical Methods*. Belmont, CA: Duxbury Press.

Selvin, Steve. 1996. *Statistical Analysis of Epidemiologic Data*, 2nd edition. New York: Oxford University.

Seyfrit, Carole L.. 1993. *Hibernia's Generation: Social Impacts of Oil Development on Adolescents in Newfoundland*. St. John's: Institute of Social and Economic Research, Memorial University of Newfoundland.

Shumway, R. H. 1988. *Applied Statistical Time Series Analysis*. Upper Saddle River, NJ: Prentice-Hall.

Stata Corporation. 2005. *Getting Started with Stata for Macintosh*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Getting Started with Stata for Unix*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Getting Started with Stata for Windows*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Mata Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Base Reference Manual* (3 volumes). College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Data Management Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Graphics Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Programming Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Longitudinal/Panel Data Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Multivariate Statistics Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Quick Reference and Index*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Survey Data Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Survival Analysis and Epidemiological Tables Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata Time-Series Reference Manual*. College Station, TX: Stata Press.

Stata Corporation. 2005. *Stata User's Guide*. College Station, TX: Stata Press.

Stine, Robert and John Fox (eds.). 1997. *Statistical Computing Environments for Social Research*. Thousand Oaks, CA: Sage Publications.

Topliss, Brenda J. 2001. "Climate variability I: A conceptual approach to ocean-atmosphere feedback." In Abstracts for AGU Chapman Conference, The North Atlantic Oscillation, Nov. 28-Dec. 1, 2000, Ourense, Spain.

Tufte, Edward R. 1997. *Visual Explanations: Images and Quantities, Evidence and Narratives*. Cheshire, CT: Graphics Press.

Tukey, John W. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Velleman, Paul F. 1982. "Applied Nonlinear Smoothing," pp. 141-177 in Samuel Leinhardt (ed.) *Sociological Methodology* 1982. San Francisco: Jossey-Bass.

Velleman, Paul F. and David C. Hoaglin. 1981. *Applications, Basics and Computing of Exploratory Data Analysis*. Boston: Wadsworth.

Ward, Sally and Susan Ault. 1990. "AIDS knowledge, fear, and safe sex practices on campus." *Sociology and Social Research* 74(3):158-161.

Werner, Al. 1990. "Lichen growth rates for the northwest coast of Spitsbergen, Svalbard." *Arctic and Alpine Research* 22(2):129-140.

World Bank. 1987. *World Development Report* 1987. New York: Oxford University.

World Resources Institute. 1993. *The* 1993 *Information Please Environmental Almanac*. Boston: Houghton Mifflin.

关键词索引